Masters
Program
in **Geospatial
Technologies**

RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA: A CASE STUDY OF TERAI DISTRICTS OF NEPAL

Tina Baidar

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

RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA: A CASE STUDY OF TERAI DISTRICTS OF NEPAL

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:
Mário Silvio Rochinha de Andrade Caetano, PhD
Associate Professor, Nova Information Management School,
Universidade Nova de Lisboa (UNL),
Lisbon, Portugal

February 21, 2020

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, **Prof. Dr. Filiberto Pla Bañón** for his persistent support and supervision, for providing full support including well-equipped room and access to the server and for treating me more like a staff of the department. I am also equally thankful to my co-supervisor, **Dr. Rubén Fernández Beltrán** for his continuous guidance, technical and moral support throughout and for cheering me up whenever needed. I consider myself lucky to get this opportunity to have worked with them. Likewise, I am thankful to my co-supervisor, **Prof. Dr. Mário Silvio Rochinha de Andrade Caetano** for his constructive feedback on the thesis dissertation.

I am grateful to **Hari Krishna Dhonju** for helping me with the thesis concept at the start and providing me valuable suggestions thereafter. I would also like to thank **Sanjeevan Shrestha** for always being there and for his support and encouragement not only during this thesis period but throughout the course. I am also grateful to **Jeevan Shrestha** and **Bishruti Siku** for their productive comments and suggestions and for being there always when in need as a family. Thank you, **Ganesh Prasad Sigdel**, for providing last-minute support in the compilation of this document.

Finally, I would like to thank **my parents and family** in Nepal for all their unconditional support and love, and **my GeoTech family** here who has become a beautiful part of this journey. I express my due regards to the **Erasmus Mundus Program** for granting me the financial support to pursue this excellent master's course. I dedicate this work to **my mom and dad**. Thank you for having faith in me.

RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA: A CASE STUDY OF TERAI DISTRICTS OF NEPAL

ABSTRACT

Crop monitoring, especially in developing countries, can improve food production, address food security issues, and support sustainable development goals. Crop type mapping and yield estimation are the two major aspects of crop monitoring that remain challenging due to the problem of timely and adequate data availability. Existing approaches rely on ground-surveys and traditional means which are time-consuming and costly. In this context, we introduce the use of freely available Sentinel-2 (S2) imagery with high spatial, spectral and temporal resolution to classify crop and estimate its yield through a deep learning approach. In particular, this study uses patch-based 2D and 3D Convolutional Neural Network (CNN) algorithms to map rice crop and predict its yield in the Terai districts of Nepal. Firstly, the study reviews the existing state-of-art technologies in this field and selects suitable CNN architectures. Secondly, the selected architectures are implemented and trained using S2 imagery, ground-truth and auxiliary data in addition for yield estimation. We also introduce a variation in the chosen 3D CNN architecture to enhance its performance in estimating rice yield. The performance of the models is validated and then evaluated using performance metrics namely overall accuracy and F1-score for classification and Root Mean Squared Error (RMSE) for yield estimation. In consistency with the existing works, the results demonstrate recommendable performance of the models with remarkable accuracy, indicating the suitability of S2 data for crop mapping and yield estimation in developing countries.

Reproducibility self-assessment (https://osf.io/j97zp/): 2, 2, 2, 1, 2 (input data, preprocessing, methods, computational environment, results).

KEYWORDS

Sentinel-2 (S2) data

Rice Crop Classification

Yield Estimation

Deep Learning

Convolutional Neural Network

ACRONYMS

Adam	Adaptive Moment Estimation.
ANN	Artificial Neural Network.
BOA	Bottom-of-Atmosphere.
CNN	Convolutional Neural Network.
Conv1D	One-dimensional CNN.
Conv3D	Three-dimensional CNN.
CSM	Crop Simulation Model.
DEM	Digital Elevation Model.
DL	Deep Learning.
DNN	Deep Neural Network.
DTL	Decision Trees.
ELU	Exponential Linear Unit.
EVI	Enhanced Vegetation Index.
FAO	Food and Agriculture Organization.
FC	Fully Connected.
FCN	Fully Convolutional Network.
FEWS	Famine Early Warning System.

FN	False Negative.
FP	False Positive.
GDP	Gross Domestic Product.
L1C	Level-1C.
L2A	Level-2A.
LSTM	Long Short Term Memory.
LSWI	Land Surface Water Index.
ME	Mean Error.
ML	Machine Learning.
MLP	Multilayer Perceptrons.
MODIS	Moderate Resolution Imaging Spectroradiometer.
MSI	Multi-Spectral Instrument.
NDVI	Normalized Difference Vegetation Index.
NIR	Near Infrared.
OLS	Ordinary Least Squares.
R&D	Research and Development.
RBF	Radial Basis Function.
ReLU	Rectified Linear Unit.
RF	Random Forest.
RGB	Red, Blue, Green.
RMSE	Root Mean Squared Error.

RNN	Recurrent Neural network.
S2	Sentinel-2.
SAFE	Standard Archive Format for Europe Format.
SAR	Synthetic aperture radar.
SDH	Scientific Data Hub.
SGD	Stochastic Gradient Descent.
SVM	Support Vector Machine.
SVR	Support Vector Regression.
SWIR	Shortwave Infrared.
TN	True Negative.
TOA	Top-of-Atmosphere.
TP	True Positive.
UAV	Unmanned Aerial Vehicle.
UTM	Universal Transverse Mercator.

INDEX OF THE TEXT

INDEX OF TABLES

INDEX OF FIGURES

1 INTRODUCTION

1.1 Contextual Background

Increasing population and climate change have imposed challenges in agriculture sector in terms of productivity, food security and sustainability [1]. The 2030 Agenda for Sustainable Development of the United Nations has explicitly defined that improving food security and ending hunger is one of their primary goals [2]. Timely and efficient agricultural monitoring is key to achieve this goal. In this context, remote sensing-based techniques are proven to be an effective technique to detect the regions with inadequate and poor crop growing conditions, to determine food-insecure areas and populations, and to monitor the development of crops [3]. Crop type mapping and yield estimation are essential for monitoring and decision-making process such as crop insurance, financial market forecasting, and addressing food security issues. Multi-temporal and multispectral satellite images can be used to identify various types of crops and monitor their growth stages. Freely available multispectral satellite sensor imagery with coarse-to-medium spatial resolution such as Moderate Resolution Imaging Spectroradiometer (MODIS) and Landsat are the most frequently used data source in optical remote sensing [4][5][6]. However, the moderate spatial resolution of MODIS and lower temporal resolution of Landsat limits the accuracy in determining a detail crop extent. In this regard, the free availability of S2 imagery with high spatial and temporal resolution has unlocked extensive opportunities for agricultural applications that include crop mapping and monitoring as well.

As a part of the Copernicus European Earth Observation program, the S2 mission offers global coverage of terrestrial surfaces by means of high-resolution multi-spectral data [7]. In particular, the S2 mission includes two identical satellites (S2A launched on 23 June 2015 and S2B followed on 7 March 2017) that incorporate the Multi-Spectral Instrument (MSI). The S2 mission offers global coverage of land surfaces with innovative wide-swath of 290km, high spatial resolution ranging from 10m to 60m, high spectral resolution with 13 bands in the visible, near infra-red and shortwave infrared of the electromagnetic spectrum and high temporal resolution with 5 days revisit frequency. With these features, S2 imagery has potential to overcome the issues with coarse satellite imagery and costly data sources. Especially in developing countries, where crop monitoring process is very important but challenging at the same time due to the major problem of data availability, the S2 mission has an extensive scope to address these challenges and gaps. [8] also indicated that high-resolution crop type maps are not available globally and S2 imagery has high potential to fill this breach of data availability and contribute to timely and accurate crop type maps.

In addition to high-quality remote sensing data, advanced and intelligent algorithms are essential for obtaining high accuracy in classification and yield estimation. Recently, deep learning algorithms such as CNN, Deep Neural Network (DNN) and Recurrent Neural network (RNN) have shown great potential in remote sensing applications such as land cover mapping, crop mapping tree species mapping and crop yield estimation. Traditional approaches for classification and regression tasks require feature engineering and field knowledge to extract features from images. While with deep learning algorithms, they have the ability to learn from multiple levels of representation [9]. These algorithms are proven to have outperformed the classical machine learning approaches In summary, the state-of-the-art on crop type classification and yield estimation has shifted from conventional machine learning algorithms to highly advanced deep learning classifiers, and from depending on only spectral features of single image to using both spectral and temporal information together for better accuracy [9].

1.2 Problem Statement and Motivation

Undoubtedly, a number of programs like Food and Agriculture Organization (FAO), Famine Early Warning System (FEWS), etc. exist that use satellite observations for crop monitoring on a regional to a global scale. Despite all these efforts, when it comes to a national or local scale actions, these monitoring systems do not effectively fulfill the need for crop monitoring and management [10]. Additionally, the use of traditional approaches for image classification is a time-consuming process that needs to be altered with the change in datasets and this hinders the timely availability of information, crucial for monitoring and decision-making process. Existing approaches for crop mapping and crop yield estimation mostly rely on survey data and other variables related to crop growth such as weather, precipitation and soil properties. These approaches are very successful in developed countries like in the United States, where adequate data with high quality are freely available [11]. However, in developing countries such as Nepal, complete and timely updated data are typically not available where the prediction of yields is most needed.

In Nepal, the national economy is dominated by agriculture. In a country where major part of the population is directly engaged in farming for their living, a high degree of spatial and temporal variability, traditional agricultural practices, climate change, and its vulnerability have imposed a serious challenge in effective and sustainable agriculture production [10]. In these circumstances, freely available S2 imagery motivates to conduct research and development in this field to address such challenges and make an effort towards sustainable solutions. The fact that Deep Learning (DL) in agriculture has outperformed the existing approaches and is growing popularity with its modern and promising technique, [12] also adds motivation to conduct this study.

1.3 Research Aims and Objectives

This work aims to support the crop monitoring process by investigating the viability of S2 data for rice crop classification and crop yield estimation in the Terai region of Nepal using a deep learning approach. To fulfill this aim, the following are the specific objectives:

- To review existing state-of-art deep learning algorithms for rice crop classification and yield estimation and select suitable architectures.
- To implement and optimize the performance of the chosen architectures.
- To evaluate the performance of classification and yield estimation models using performance metrics and validate their performance with reference to the existing works.

1.4 Methodology

The structure of this thesis can be broadly organized into four stages namely; i) review and choice of network architecture; ii) dataset download/collection and preparation; iii) design, implementation, and analysis of network architectures, and iv) performance comparison. The detailed structure of the overall methodology is shown in 1.1.

In the first stage, a number of existing deep learning approaches used for crop classification and crop yield estimation were reviewed. Considering the availability of the dataset and performance of deep learning algorithms, DNN was chosen as the core algorithm for addressing both the problems of classification and yield estimation. Besides this, Support Vector Machine (SVM) and Support Vector Regression (SVR), the mostly used classical machine learning algorithms for classification and regression problems, were used as baseline algorithms to compare the results of chosen CNN architectures against them.

Secondly, considering the rice crop phenology in the study area i.e., the Terai region of Nepal, S2 Level-1C (L1C) MSI images were downloaded for the years 2016-2018. These images were converted to atmospherically corrected S2 products, resampled to 20 m and further processed to feed the images to the CNN architecture for rice crop classification. In addition to these processed S2 images, ground truth-rice yield data, soil and climate data were collected and processed to 20 m in order to implement CNN architecture for yield estimation.

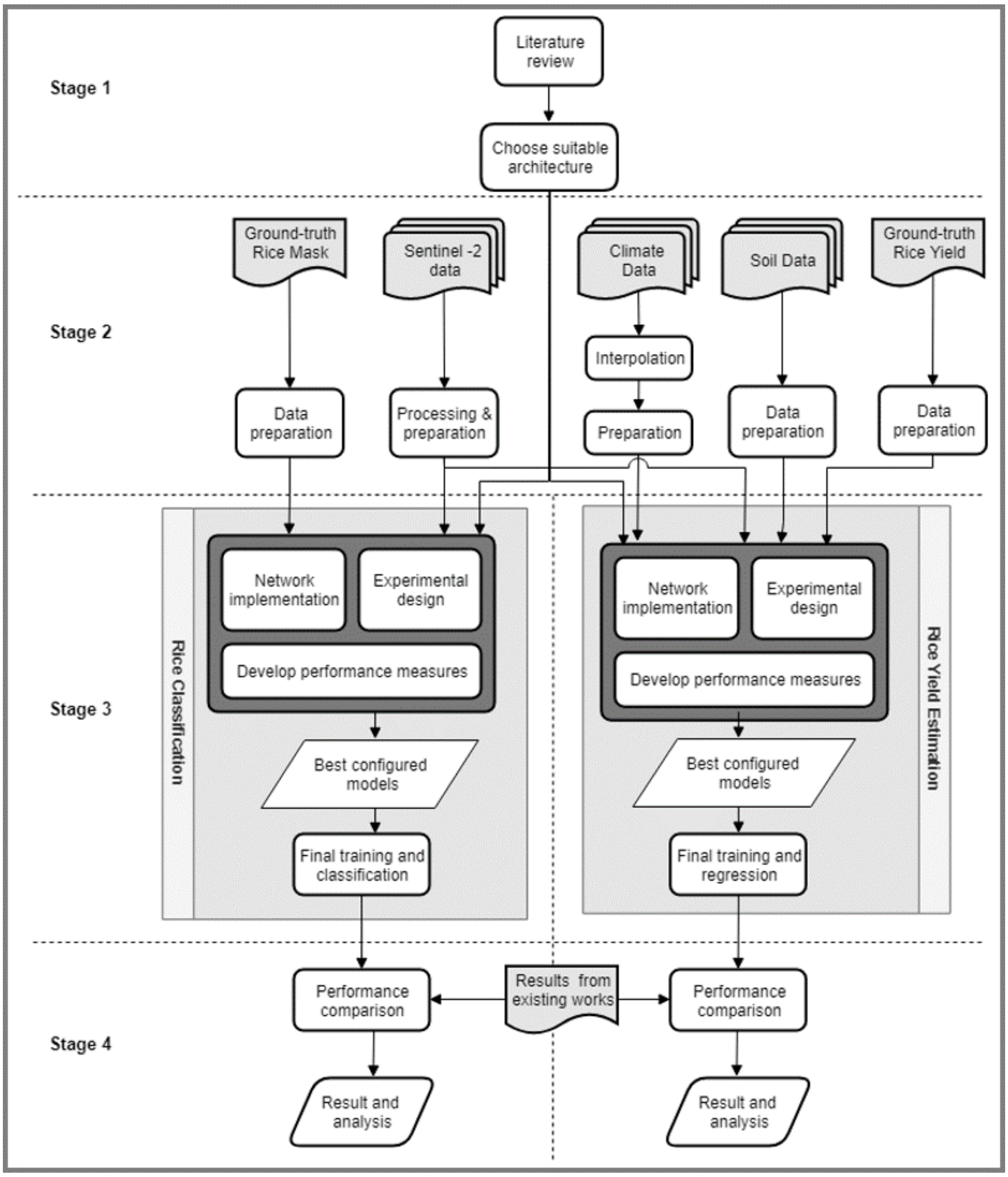

Figure 1.1: Overall methodology of the thesis

The third stage deals with the design, implementation, and analysis of the chosen CNN architectures. For classification purpose, three variations of 2D CNN and one 3D CNN architectures were implemented to inspect how the performance of the classifiers vary with the variations in the network architecture. For the second case which is the yield estimation process, two CNN architectures (2D and 3D CNN) were designed and employed. Several experimental designs were conducted so as to select the optimal values of the hyperparameters.

Finally, in the last phase, the performance of all the models using quantitative and qualitative approaches was evaluated. For a quantitative approach, two metrics namely overall accuracy and F1-score were used as measures to evaluate the performance of each classification approach. Similarly, RMSE was used for the performance evaluation of architectures used for yield estimation. Additionally, visual inspection of classification and yield estimation maps were used as a means for qualitative evaluation. Lastly, the optimized performance of the chosen architectures was validated with reference to the corresponding works from which these architectures were adopted to verify the usability of S2 data and the approach in developing countries like Nepal.

1.5 Contribution

In a broader picture, this study is one of the first studies (as per the author's knowledge) that has introduced the use of S2 data for rice yield estimation including its classification by using a deep learning approach in developing countries with a case study in Nepal. The main contributions of this thesis consist of:

- Exploring the feasibility of using Sentinel 2-MSI data for crop classification and yield estimation in developing countries.
- Validating the suitability of S2 data combined with a deep learning approach for accurate rice crop management and sustainability with a case study of Nepal.
- Comparison of chosen CNN architectures and finding the best performing model in the study area.
- Introducing a variation in existing 3D CNN architecture [13] by removing the channel compression module to optimize its performance.

1.6 Thesis Organization

This thesis is divided into seven chapters. Chapter 1 introduces the contextual background of the thesis, states the problem and motivation behind the work, aims and objectives and highlights the contribution of the work. Chapter 2 reviews the related works on the existing methods for crop classification and yield estimation using remote sensing imagery and emphasizes on the state-of-art technologies with special focus on rice crop. Also, this chapter deals with the selected CNN architectures to be implemented to fulfill the objective of the study. Chapter 3 provides the theoretical background of the convolutional neural network architecture along with the baseline algorithms. Chapter 4 familiarizes with the study area and presents the datasets used as input for CNN architectures. Chapter 5 presents the methodological description of implementation, training, experimental settings and performance evaluation of chosen CNN-based classifiers for classification and yield estimation purposes. In Chapter 6, the results of the experiments are shown, interpreted and discussed. Finally, the thesis ends with the conclusion and future directions of the work in Chapter 7.

2 LITERATURE REVIEW

This chapter provides a comprehensive review of the existing state-of-art on crop classification and yield estimation using remote sensing technology combined with deep learning techniques. The chapter is divided into two sections. The section 2.1.1 represents the traditional approaches used for crop mapping, especially rice crops. Thereafter, it deals with the deep learning approaches used in classification and finally explains the choice of network architectures and baseline. Similarly, the section 2.2.1 explains the traditional approaches used, deep learning algorithms, and choice of architecture and baseline for (rice) crop yield estimation.

2.1 Crop Classification

2.1.1 Traditional Approaches for Crop Classification

Remote sensing-based techniques are proven to be an effective technique for crop classification and crop area estimation compared to traditional ground-based surveys which consume a lot of money and time. Time-series observations are essential to monitor the crop growth and multi-temporal remote sensing is an efficient source for this. Its classification accuracy is higher than using mono-temporal images because it considers the crop information in different growth stages [4][14]. In multi-temporal remote sensing, various classification approaches and data sources have been used in past studies for crop mapping, especially rice crop. In terms of approaches for rice crop classification, they can be grouped into two types in general which are supervised and unsupervised classification [15]. Under supervised classification, knowledge-based [16] and phenology-based [17][18] approaches are the typical methods used.

In terms of data source, both optical and micro-wave based remote sensing have been used. In optical remote sensing, vegetation indices derived from images are used as a basis for rice crop classification. The commonly used indices are Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), and Land Surface Water Index (LSWI) [9][19]. Direct use of time series vegetation indices is effective for the crops that show distinct temporal characteristics [4][16]. A widely used approach of processing multi-temporal data is extracting the phenological metrics which may improve the classification accuracy than using the original vegetation indices values. Freely available multispectral satellite sensor imagery with a coarse-to-medium spatial resolution like MODIS and Landsat are the most frequently used data source in optical remote sensing [4][5][6][10]. However, the moderate spatial resolution of MODIS limits the accuracy in determining a crop extent and lower temporal resolution of Landsat images limits the usage of phenology-based crop classification and monitoring. A major limitation found in optical remote sensing is the influence of cloud and cloud shadows during the rice growing season. Considering this aspect, microwave-based remote sensing has an advantage that it can work in all weather conditions. However, low temporal resolution and data cost have limited agricultural applications of Synthetic aperture radar (SAR) images.

Recently, the free availability of Sentinel imagery with the high spatial and temporal resolution has opened a broad opportunity for a wide range of pre-operational and operational applications in the agricultural domain [20] resolving the issues with the coarse resolution of optical imagery and SAR. [21] has demonstrated Sentinel -1A and 2A's remarkable potential for crop classification and suggested the integration of Sentinel-1A and 2A for high accuracy crop classification. However, with only Sentinel-1, [19] showed that high heterogeneity in pixel values of SAR image (Sentinel-S1) lowered the accuracy of land cover classification. It also concluded that although combining both sentinel-1 and sentinel-2 resulted in the highest accuracy, when multi-temporal data is available, it is not necessary to combine images from different sensors to obtain high accuracy results.

The use of traditional approaches for image classification is a time-consuming process that needs to be altered with the change in data sets and this hinders the timely availability of information, crucial for monitoring and decision-making process. In addition to high-quality remote sensing data, advanced and intelligent classification algorithms are essential for obtaining high accuracy n classification provided that there are enough training data. Conventional classification methods such as SVM [22] Random Forest (RF) [23], and Decision Trees (DTL) [17][21][24][25][26] have been successfully applied in crop classification. While significant effort has been done to classify crops with these approaches, these algorithms require careful feature engineering and considerable domain knowledge to extract features from raw data [9][20].

2.1.2 Deep Learning Algorithms for Crop Classification

Recent developments in neural network methods through more layers and back-propagation optimization (deep neural network) have significantly increased the use of neural networks that have been developed several years ago. DL in agriculture is new and gaining momentum recently. Based on the study of 40 recent (past four years) research papers that used DL in agriculture, [12] have concluded that DL has outperformed traditional approaches for image classification. DL is a deeper neural network that provides a hierarchical representation of the data which allows better learning capabilities in terms of capturing the full complexity of problems to be addressed. In this section, promising DL approaches used for crop classification are discussed.

One of the first works on crop classification using CNN classifiers to multi-source multi-temporal satellite imagery was published by [20]. The study introduced one 2D CNN for spatial feature learning and one 1D CNN for spectral feature learning for crop classification which outperformed Multilayer Perceptrons (MLP) and RF classifier with overall classification accuracy of 94.6% achieved by 2D CNN. [27] employed a patch-based deep-learning CNN algorithm to extract rice crop using multi-temporal Landsat-like data, phenology data, and land-surface temperature. With multitemporal spectral bands as input, the 2D CNN achieved the overall classification accuracy of 91.23% demonstrating its better performance as compared to traditional machine learning classifiers, support vector machine and random forest. However, the paper did not consider the spatial pattern of the study area.

Authors in [27] highlighted that with a 2D CNN classifier, the extracted features in the third dimension (spectral or temporal) are averaged or collapsed to a scalar which results in inaccurate extraction of third-dimensional features. To overcome this problem of 2D CNN, the study proposed a 3D CNN with an active learning strategy for crop classification including rice using multi-temporal satellite data. When compared to a 2D CNN that shared the same architecture of 3D CNN except for the learned representation, 1.2% increase in overall accuracy was seen concluding that 3D CNN could be a better feature extractor for spatio-temporal remote sensing data. [28] used two deep neural networks: one-dimensional CNN (Conv1D) and Long Short Term Memory (LSTM) based RNN for multi-temporal crop classification using Landsat EVI time series. Three widely used non-deep-learning classifiers namely XGBoost, RF, and Support Vector Machine (SVM) were tested for comparison. The study demonstrated the lowest accuracy of LSTM with an overall accuracy of 82.41% among all classifiers and One-dimensional CNN (Conv1D) achieved the highest accuracy of 85.54% . The paper also attempted Three-dimensional CNN (Conv3D), which had the highest accuracy among all since it utilized pattern in both spatial and temporal dimensions, providing more scope of classification with 3D CNN for higher accuracy.

2.1.3 Choice of Network Architecture and Baseline

As this study focuses on assessing the performance of existing deep learning classifiers with S2 imagery in the study area, the choice of network architecture is based on the review of previous works as discussed in 2.1.2. Exploring the trend of deep learning in rice crop classification, it is confirmed that CNN can perform better than other algorithms including LSTM for rice crop classification. Therefore, CNN is chosen as a core algorithm for rice crop mapping. [27] employed a simple patch-based ConvNet network with reference to the network architecture used by [29] for rice crop mapping and demonstrated its highest accuracy than the two full CNNs, a patch-based VGG-16 network [30] and a pixel-based

S.N.	Models	Reference Paper
1.	ConvNet (2D CNN)	Zhang et al. [27]
2.	2D CNN	Ji et al. [33]
3.	3D CNN	Ji et al. [33]
4.	SVM Baseline	

Table 2.1: List of chosen patch-based CNN architectures and baseline algorithm

Fully Convolutional Network (FCN) [31]. In the same year, [27] proposed a 3D CNN approach for rice crop classification which was based on widely used neural network architecture, VGGnet [32] where all 2D convolutions are replaced by 3D convolutions. The same architecture was employed to develop 2D CNN architecture and the performance evaluation of these two models demonstrated the higher classification accuracy of 3D CNN. Considering the higher accuracy of patch-based CNN than pixel-based, both the above-mentioned architectures were employed with an objective of evaluating and comparing the classification accuracy of the selected CNN architectures with S2 data in the study area.

An SVM classifier, well-known for high performance among the non-deep learning classifiers [27] and often established as a baseline model in the classification tasks is used as a baseline model in this study. In general, SVM with Radial Basis Function (RBF) kernel is considered to be a reasonable first choice to handle the case where the relationship between class labels and attributes is nonlinear. Having said that, [29] demonstrated higher accuracy of linear SVM in comparison to SVM with RBF kernel when the input dataset is huge. Therefore, taking this into, a linear SVM has been employed as a baseline model for comparing classification results. The network architectures and baseline algorithm chosen for rice classification purpose are summarized in Table 2.1.

2.2 Crop Yield Estimation

2.2.1 Traditional Approaches for Crop Yield Estimation

Establishing models for estimating crop yield is very important but challenging at the same time in the remote sensing field. The classical approaches to predict or estimate crop yield include the use of manual surveys, Crop Simulation Model (CSM) or remote sensing data. Manual surveys require in-situ crop information, which are costly and difficult to scale to other regions. CSMs simulate the crop development, growth and yield through statistical, mechanistic or functional models as functions of soil, weather and management practices [34]. However, these models require large datasets for model calibration and therefore may not be applicable for developing countries where data is scarce and sparse. Many studies have shown the use of remote sensing technology to estimate the crop production by employing statistical methods such as regression model. [35] employed a piecewise linear regression method with a breakpoint to predict corn and soybean yields using NDVI, surface temperature, precipitation, and soil moisture. [36] used a stepwise regression method for estimating winter wheat yields using MODIS NDVI data. [37] estimated crop yields with the prediction error of about 10% in the US Midwest by employing Ordinary Least Squares (OLS) regression model using time-series MODIS products and climate dataset. In summary, the classical approach for crop yield estimation is mostly based on the multivariate regression analysis using the relationship between crop yields and agro-environmental variables like vegetation indices, climatic variables, and soil moisture. These approaches have compactly summarized the information related to vegetation growth with the use of vegetation indices which depend on a smaller number of available bands, usually two. But the bands which are ignored with this approach could have additional important information to predict the yield with more accuracy.

With advances in Machine Learning (ML), researchers have applied machine learning algorithms to remote sensing imagery for crop yield prediction. [38] introduced the use of Artificial Neural Network (ANN) through a back-propagation algorithm in forecasting winter wheat crop yield using remote sensing data to overcome the problem that existed in using traditional statistical algorithms (especially regression models) due to nonlinear character of agricultural ecosystems. The study demonstrated high accuracy of ANN compared to results from a multi-regression linear model (MR model). The commonly used machine learning techniques include SVM, DT, and MLP [39]. These techniques have contributed to improving the accuracy of crop prediction. However, these approaches require feature engineering. On the other hand, with deep learning algorithms, they automatically learn the relevant features from the raw data effectively.

2.2.2 Deep learning in Crop Yield Estimation

One of the first works on employing a deep learning approach for crop yield estimation was done by [40]. The study employed a Caffe-based deep learning regression model with satellite, climate and environmental data to estimate corn crop yield at county-level in the United States. The proposed architecture achieved an RMSE of 6.298 outperforming the performance of SVR. [11] introduced a new dimensionality reduction technique i.e. treating raw images as histogram of pixel counts under the assumption of permutation variance and trained deep learning architectures CNN and LSTM on these histograms to predict county-level soybean yields in U.S. To address spatial and temporal dependencies across data points, the paper proposed the use of linear Gaussian Process layer on the top of these neural network architectures. To compare with prior works, the inputs datasets used for this study are long-term (2003-2015) MODIS satellite-based surface reflectance, LST and land cover data. The baseline methods used to compare the results were ridge regression, decision trees, and a DNN with 3 hidden layers. The results demonstrated that CNN and LSTM approaches outperformed the other competing techniques used as baselines. Also, prediction was improved by addition of linear GP resulting in 30% reduction of RMSE from baselines.

The location invariant assumption by [11] discards the spatial information of satellite imagery which can also be crucial information for crop yield estimation. To overcome this, [13] introduced a 3D CNN approach that considers both spatial and temporal features for yield prediction. Firstly, the paper replicated the Histogram CNN approach of [11] with the same input dataset and set this as a baseline for their novel approach, 3D CNN. Considering the computational cost, a channel compression model was applied to lessen the channel dimension from 10 to 3. Thereafter a 3D CNN was stacked to the model. The proposed 3D CNN architecture outperformed the replicated Histogram CNN and non-deep learning classifiers used as baselines [11] with an average RMSE of 5.27 bushels per acre (around 355 kg/ha). Continuing to the same data, [41] proposed a novel approach of deep CNN-LSTM model for end-of-season and in-season soybean yield prediction. The proposed network consisted of 2D CNN followed by LSTM where CNN learns the spatial features and LSTM is used to learn the temporal features extracted by CNN. The proposed model achieved reduced RMSE of average 329.53 kg/ha which was better than CNN and LSTM models.

Recently, [42] applied a CNN model to predict crop yield using NDVI and Red, Blue, Green (RGB) images acquired from Unmanned Aerial Vehicle (UAV). The result showed the better performance of CNN architecture with RGB data. [43] proposed a novel CNN architecture which used two separate branches to process RGB and multispectral images from UAV to predict rice yield in Southern China. The resulting accuracy outperformed the traditional vegetation index-based regression model. The study also highlighted that unlike the vegetation index-based regression model, the performance of CNN for yield estimation at the maturing stage is much better and more robust.

2.2.3 Choice of Network Architecture and Baseline

Based on the related literature discussed in section 2.2.2, CNN and LSTM are mostly used deep learning algorithms to address the problem for crop yield prediction with high accuracy. On the one hand, when these algorithms were used separately, [11] demonstrated that CNN achieved higher accuracy than LSTM. On the other hand, the combined model, CNN-LST performed relatively better [41]. However, these experiments were conducted using a long-term dataset with a high temporal dimension. The limited temporal dimension of the dataset in this study limits the use of the LSTM model. Therefore, for rice crop yield estimation also, CNN is chosen as a core algorithm. Exploring the performance of various architectures of CNN in the previous studies, **3DCNN** proposed by **[13]** achieved the highest accuracy to date and is therefore chosen as the base architecture for this study. Similar to SVM, **SVR** is commonly used to perform yield prediction and will be used as a baseline to compare the performance of 3D CNN against it.

3 THEORETICAL BACKGROUND

This chapter is meant to serve as the theoretical foundation of the concepts used in the thesis. The first section presents a brief explanation about the classical machine learning approaches which have been selected as baselines for the crop classification and yield estimation tasks. The second section starts with some limelight on deep learning and neural networks which is followed by background concepts of CNN in detail.

3.1 Classical Machine Learning Approach

This section gives a brief theoretical overview of the classical machine learning algorithms namely SVM and SVR used as baselines for comparing the performance of CNN algorithms used for rice crop classification and yield estimation respectively.

3.1.1 Support Vector Machine

SVM, as proposed in statistical learning theory [44], is known to be an effective kernel-based classification algorithm that is based on statistical learning theorem. The main objective of SVM is to find the optimal linearly separating hyperplane which maximizes the margin. The key idea behind kernel tricks is to map data in to a higher-dimensional space so that different groups or classes can be linearly separable [45]. Mathematically, the optimization problem for support vector machine can be formulated as,

$$\begin{gathered} \min_{w,b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y_i \left(w^T x_i + b \right) \geq 1 \text{ }, i = 1, ..., m \end{gathered} \tag{3.1}$$

where (w, b) defines a hyperplane that separates all the training data into the two labeled classes, x_i are the input data and y_i are labels. This is a quadratic optimization problem subject to linear constraints. The objective is a convex function with a unique minimum. For linearly non-separable classes, non-linear decision boundaries can be constructed using for instance, a radial basis function (RBF) [16][46].

3.1.2 Support Vector Regression

A regression model estimates a continuous-valued multivariate function. SVR is an extension of SVM algorithm which was introduced for regression scenarios to predict the numerical property values [47]. In case of SVM an optimal hyperplane is generated for class label prediction, on the other hand in SVR, a different function is derived based on training data to predict numerical values. SVR has proven to be an effective tool for estimating the real-value function.

3.2 Artificial Neural Networks and Deep Learning

ANN and DL are state-of-art technology that is providing the best solution to existing machine learning algorithms. ANN is a type of machine learning and is inspired by the biological nervous system and consists of interconnected neurons that work in a distributed fashion to learn from input in order to optimize its final output [48]. DL also belongs to a broader family of machine learning and is inspired by ANN. Although DL is similar to ANN, it constitutes a deeper neural network that allows better learning capabilities to capture the full complexity of the considered data or phenomenon [49]. Deep learning architectures such as deep neural networks, unsupervised pre-trained networks, recurrent neural networks, recursive neural networks and CNN have been successfully applied to diverse fields including computer vision, speech recognition, audio recognition, natural language processing, medical image analysis, and so on with high accuracy and have proven to be superior than the classical machine learning algorithms. Since we have chosen CNN as a core algorithm for this study, the following subsections deal with the detailed architecture of CNN.

3.2.1 Convolutional Neural Networks

CNNs are biologically inspired feed-forward neural networks that extend the classical artificial neural network approach by adding multiple convolutional layers and filters that allow representing the input data in a hierarchical way [50]. CNN exhibits high performance in image processing tasks, thereby positioning itself as the current state-of-the-art of image classification methods. Traditional neural networks (deep or shallow ones) which are characterized by 1D architectures are composed of (Fully Connected (FC)) layers. Whereas in CNN, hidden activation is calculated by multiplying small local inputs against weights and exploits spatially local association by applying a local pattern of connectivity between adjacent layer neurons. The weights are then shared across the entire input space. The output volume of CNN is composed of feature maps which are then used as input to the next layer. The basic concepts employed in CNN are explained in the following subsections:

3.2.1.1 Basic Architecture

The basic architecture of CNN consists of alternatively stacked convolutions blocks and pooling layers followed by one or more fully connected layers. The convolutional blocks are usually composed of convolutional layers followed by batch normalization layers and nonlinear activation functions. The convolution and pooling layers act as feature extractors from the input images. At last, the fully connected layer, after a series of convolution and pooling, gives the class score of each pixel through the network in a feed-forward manner. A detailed explanation is presented here.

Convolutional layer

The convolutional layer is the core building block of the convolutional network that consists of filters (also termed as kernels) which are applied over the image to extract different features learned by the network. Convolution layer computes the output of neurons that are connected to the local regions in the input by conducting dot product between their weights and biases and a specific region to which they are related in the input range [48]. Mathematically,

$$x_{l+1} = f\left(W_l x_l + b_l\right), \tag{3.2}$$

where x_{l+1} is the output with n feature maps of the l^{th} convolution layer, W_l is weight matrix defined by the filter bank with kernel size $N \times N$ and b_l is the bias of the l^{th} convolution layer, and $f\,()$ being the nonlinear activation function.

In this study, the input is taken from multispectral-temporal images. The basic way is to treat each channel independently and generate a uniform-sized image patch around the same images. Let us suppose a multispectral image $X \in \mathbb{R}^{(M \times D \times H)}$ where M, D, H are the spectral bands, width, and height respectively. The pixel $\mathrm{x}(i,j)$ of X (with $i = 1, 2, \ldots, D$ and $j = 1, 2, \ldots, H$) can be defined as the spectral vector $\mathrm{x}(i,j) \in \mathbb{R}^M = [x_1(i,j), x_2(i,j), ..., x_M(i,j)]$. Let us define a neighboring region (image patch) $q_{(i,j)} \in \mathbb{R}^{(p \times p)}$ around $x_{(i,j)}$, composed of pixels from $((i-(p2), j-(p2)), ((i+(p2), j-(p2)), ((i-(p2), j+(p2)), ((i+(p2), j+(p2))$. With q taking account of spectral information, it can be redefined as $q_{(i,j)} \in \mathbb{R}^{(M \times p \times p)}$. The convolution layer takes a $p \times p$ image patch with M channels centered at a pixel $\mathrm{x}(i,j)$ and two-dimensional filter kernel $N \times N$ with k number of filter. Let $y_k(i,j)$ be pixel value at output feature map and $w_k(r,s)$ be weight value at (r,s) at k^{th} filter. Then mathematically, the convolution process is defined as [33],

$$y_k\left(i,j\right) = \sum_{m=0}^{M} \left\{ \sum_{r=0}^{N-1} \sum_{s=0}^{N-1} w_k\left(r,s\right) x_m\left(i,j\right)\left(i+r, j+s\right) \right\} + b_k, \tag{3.3}$$

The convolution operation in a normal image down-samples the output image size by an amount that depends on the filter size. To avoid this, **padding** is used which essentially preserves the original input image size. Padding is simply a process of adding layers of zeroes to the input images. There are two types of padding namely *valid* padding which implies no padding at all and *same* padding which means the output feature map has the same dimension as the input image.

Batch normalization is a process to reduce the co-variance shift by normalizing the layer's inputs over a mini-batch. It enables independent learning process in each layer and regularizes and accelerates the training process. Mathematically, it is defined as:

$$BN\left(x\right) = \frac{x - E\left(x\right)}{\sqrt{Var\left(x\right) + \varepsilon}}\gamma + \beta, \tag{3.4}$$

where E is expectation operator, γ and β are learn-able parameter vectors, respectively, and ε is a parameter for numerical stability. This layer makes the hyperparameter search much easier and make the neural network more robust and enable easier training of the deep network.

Activation functions are used to embed non-linearity into the neural network thereby enabling the neural network to learn nonlinear representations. Activation function can be expressed as:

$$Z\left(y_k\left(i,j\right)\right) = f\left(\sum_{k=1}^{K} x_k\left(i,j\right)w_k + b_k\right), \tag{3.5}$$

where $f\left(\right)$ is a non- linear function. There are varieties of activation functions such as sigmoid function, Tanh, Rectified Linear Unit (ReLU) [51] and Exponential Linear Unit (ELU). Sigmoid and Tanh tend to saturate when initialized weights are too high and give rise to a problem of vanishing gradient if gradient tends to zero. ReLU solves this problem by thresholding the negative inputs to zero and passing the positive inputs unchanged [52]. ReLU is proven to be computationally efficient and effective for convergence and is defined as

$$A\left(y_k\left(i,j\right)\right) = max\left(0, Z\left(y_k\left(i,j\right)\right)\right), \tag{3.6}$$

Pooling layer

The pooling layer is a sub-sampling operation along the spatial dimensions of feature maps, typically applied after a convolution layer, which does some spatial invariance [48]. Usually, in pooling, some predefined functions (e.g. maximum, average, etc.) are applied to summarize the signal and spatially preserving discriminant information. **Max-pooling** is a non-linear sub-sampling operation whereas **average pooling**, on the other hand, can be thought of as a low-pass (averaging) filter followed by sub-sampling. The pooling region can be overlapped or non-overlapped and in the latter case, the more information is lost. The output of max pooling for a local region of dimension $k \times k$ can be mathematically defined simply as,

$$y_k\left(i_p,j_p\right)=\max_{0\leq i_p\leq n_p-1,0\leq j_p\leq n_p-1} A\left(y_k\left(i,j\right)\right), \tag{3.7}$$

The parameters to be defined by the user are the size of the window and stride (number of steps between the consecutive convolutions). After pooling layer the output is a 3D cube composed of one 2D map per filter with a reduced spatial dimension.

Fully Connected Layer

In a typical CNN architecture, the FC layer is usually placed at the end of the network after several convolutions and pooling layers. FC layer takes the output of the previous layers, flattens them and turns them into *n (number of classes)* dimensional vector that can be input to the next stage. Each number in this *n*-dimensional vector represents the probability that a certain feature belongs to one of the predefined classes. If there are multiple FC layers, the initial layers take the outputs from previous layers and apply weight to predict the correct label and the last FC layer gives the final probabilities for each label. In a FC layer, neurons have connections to all the activations in the preceding layers which is analogous to the output layer of MLP [48].

Classification layer

The output from the fully connected layer is fed to the classifier layer to calculate the probabilistic output of each class. The classification layer calculates the cross-entropy loss for multi-class classification problems with mutually exclusive classes. The most common classifier for multi-class classification is softmax [53] and for a binary classification problem, it reduces to be a logistic regression. A Softmax function limits the output into the range of [0 1] which allows being interpreted as a probability.

Let us assume $n_k \times n_k \times U$ is the form of the output of the CNN, where U is the number of channels of output image patch m. $X = [x_1, x_2, \dots, x_u]^T$ represent the pixel value in the output of fully connected layer and softmax function is used to generate a vector $K = [k_1, k_2, \dots k_u]^T$ of real values in the range of [0,1] which represents a categorical distribution . The equation 3.8 shows how softmax function predicts the probabilities of j^{th} class given the sample vector X.

$$K_{w,b}=\frac{exp\left(xw_j\right)}{\sum_{u=1}^{U} exp\left(xw_u\right)}, \tag{3.8}$$

Regression Layer

For regression problems, the classification layer of CNN is replaced by the regression layer. A regression layer returns the regression output of the neural network and computes the half-mean-squared-loss for regression problems. Mathematically,

$$loss=\frac{1}{2}\sum_{i=1}^{n_k n_k U}\left(y_i-x_i\right)^2, \tag{3.9}$$

where $n_k \times n_k \times U$ is the dimension of final output of the CNN; y_i being target output and x_i being network prediction for response i.

Regularization

Over-fitting is a serious problem in the deep learning networks where networks are powerful enough to fit itself to the training data resulting in large gap between the training and test errors. Regularization techniques are used to prevent overfitting the data (to reduce high variance) in the network and reduce its generalization error. There are several regularization techniques such as L2 and L1 regularization, dropout, and early stopping. L1 regularization makes the model sparse and that only contributes to regularize the model to less extent and therefore is not used often. L2 regularization (also known as weight decay) is one of the most commonly used regularization techniques. It basically minimizes the sum of the square of the differences (S) between the target value $y_{i,j}$ and the estimated values $f\left(x_{i,j}\right)$:

$$S = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(y_{i,j} - f\left(x_{i,j}\right)\right)^2, \tag{3.10}$$

Dropout is another powerful regularization technique that randomly knocks out units in the network resulting in a smaller network. The smaller network seems to have a regularization effect. Early stopping is a simple yet effective regularization technique while training the neural networks. It is used to stop the training of the neural network (with training dataset) at a point when the performance on a validation dataset starts to degrade.

3.2.1.2 Training

The training process of CNN can be divided into three major steps: forward computation, loss optimization, and back-propagation and parameter updating.

Forward Computation

Firstly, the input is fed through the neural network architecture consisting of a series of convolution, pooling and fully connected layers as explained in the previous subsection. The network outputs the predicted labels or values depending upon the network architecture (classification or regression).

Loss Optimization

Secondly, the output of the network needs to be optimized by adjusting the values of parameters such as weights and bias, that are being learned by the network. The optimization problem defines the uncertainty in determining the optimal set of parameters which is quantified by the loss function. In the case of softmax classifier, the cross-entropy loss for each vector is computed as negative log-likelihood of the training dataset N under the model.

$$L\left(W,b\right) = -\frac{1}{N}\sum_{i=1}^{N}\left(y^{i}logK_{w,b}\left(x^{i}\right)\right), \tag{3.11}$$

where y^i represents a possible class and x^i is the data of i instance, W is the weights, and N represents a total number of instances.

Back Propagation

Finally, training of the network must be done for extracting the parameters that minimize the loss. The network tries to reduce this error by changing the weights of neurons in every iteration through the backpropagation mechanism. Several optimization algorithms exist to make the training process faster such as Stochastic Gradient Descent (SGD) with momentum, Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (Adam) among which we will discuss in brief about two algorithms which have been used for this study namely SGD with momentum and Adam.

Stochastic Gradient Descent with Momentum:

SGD with momentum basically calculates the exponentially weighted average of the gradients and consider that gradient to update the weights. It is used for faster convergence of the loss function. α (learning rate) and β (momentum) are the two hyperparameters which control the exponentially weighted average. In practice, the most common value of β is 0.9. Mathematically, the SGD method with momentum can be described by the following:

$$W^{(n+1)} = W^{(n)} - \Delta W^{(n+1)}, \tag{3.12}$$

where $W^{(n)}$ and $W^{(n+1)}$ denote the old parameters and new parameters respectively and $\Delta W^{(n+1)}$ represents the increment in the current iteration which is the combination of old parameter, gradient and historical increment:

$$\Delta W^{(n+1)} = \alpha\left(d_w W^{(n)} + \frac{\partial L\left(Wb\right)}{\partial W^{(n)}}\right) + \beta\Delta W^{(n)}, \tag{3.13}$$

where L(W,b) is the loss function, α is the learning rate for step length control and d_w and β denote the weight decay and momentum respectively-

Adam optimization algorithm:

Adaptive Moment Estimation (Adam), introduced by [54], is also based on mini-batch gradient descent and computes adaptive learning rates for each parameter. It is an extension to stochastic gradient descent which basically takes momentum and RMSprop and puts them together. The algorithm specifically calculates an exponential moving average of the gradient and the square gradient, and the β_1 and β_2 parameters control the decay rates of these moving averages.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_1, \tag{3.14}$$

$$\vartheta_t = \beta_2 \vartheta_{t-1} + (1 - \beta_2)\, g_t^2, \tag{3.15}$$

where m_t and ϑ_t are estimates of the first moment and second moment of the gradient respectively. In practice, the default value for β_1 and β_2 , 0.9 and 0.999 are most commonly in use. The bias corrected first, and second moment estimates are then computed as:

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t} \,, \tag{3.16}$$

$$\vartheta_t = \beta_2 \vartheta_{t-1} + (1 - \beta_2)\, g_t^2 \,, \tag{3.17}$$

Finally, these biased corrected moment estimates are used to update parameters using adam update rule:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\widehat{\vartheta}_t + \epsilon}} \widehat{m_t} \,, \tag{3.18}$$

3.2.1.3 Parameters and Hyperparameters

Parameters are the variables that model updates during the back-propagation phase. Weights and biases are the core parameters of deep neural networks. Whereas, hyperparameters are the specific “higher-level” properties of the models which should be fixed prior to the training process. The value of these parameters cannot be directly learned from the regular training process of the model. Hyperparameters need to be tuned for each problem because the best model hyperparameters for one dataset will not be best for all datasets. The process of finding the combination of hyperparameter values for a model that performs the best as measured on a validation dataset is termed as hyperparameter tuning (also known as hyperparameter optimization). Hyperparameters can be divided into two types: hyperparameters that determine the network structure such as kernel size, stride, padding hidden layers, and activation function and hyperparameters that determined the network training process such as learning rate, batch size, number of epochs, regularization techniques and so on. Here, we will focus on the second type and a brief explanation of some influential hyperparameters is presented in the following subsection:

Learning Rate:

Learning rate simply means how fast a network learns. It quantifies the learning process of a model that can be used to optimize its capacity. Choosing a learning rate is challenging as a value too small causes the model to take much time to converge, whereas a value too large cause model to diverge and loss might fluctuate indefinitely [55]. A general approach used by many of the previous works was to start out with a high learning rate and lower it as the training goes on. One point to note is that the learning rate is very dependent on the network architecture. The updating of the learning rate can be formulated as:

$$\alpha_t = \alpha_0 \; \forall t < \tau \tag{3.19}$$

$$\alpha_t = \alpha_0 t^{dt}, \tag{3.20}$$

where α_0 and α_t are initial learning rate and learning rate at iteration t; d is decay parameter. τ and α are set up to adapt depending upon the present thresholds of the loss function.

Mini-batch size:

When the mini-batch size is equal the size of the input, then it is known as batch gradient descent (too long per iteration) and when the mini-batch size is equal to the one (noisy and doesn't converge), then it is known as Stochastic gradient descent. In practice, the mini-batch size of an appropriate value between 1 and the max value (not so big or small) is chosen which gives the fastest learning.

Number of epochs:

It is a hyperparameter that controls the number of complete passes through the training dataset. The weights are updated after each epoch and hence produce better results. However, using many epochs might over fit the training process and this is when early stopping can be used which prevents the overfitting of the model.

3.2.2 3D Convolutional Neural Network

In 2D CNN, all the spatio-temporal images are stacked together as the input and after the first convolution, the temporal information is collapsed completely [33]. To prevent this, 3D CNN uses 3D convolutions that apply 3-dimensional filters to the dataset resulting in a 3-dimensional volume space. In 3D CNN, the 3D convolution pooling operations are performed spatio-temporally which shows its ability to model the temporal information better than 2D CNN [56]. 3D CNN is not only limited to event detection in videos or 3D medical imaging but also applied to 2D input space such as images.

The convolution in equation 3.3 for 2D convolution can be modified to extract and reserve the dynamic features through consecutive periods and can be written as:

$$y_k(i,j,h) = \sum_{n=0}^{M} \left\{ \sum_{r=0}^{N-1} \sum_{s=0}^{N-1} \sum_{t=0}^{N-1} w_k(r,s,t)\, x_m(i,j,h)\,(i+r, j+s, h+t) \right\} + b_k, \tag{3.21}$$

where $x_m(i,j,h)$ is input pixel at input image or feature map and $y_m(i,j,h)$ are input and output pixel value at output feature map, $w_k(r,s,t)$ be weight value at (r,s,t) at k^{th} filter and b is the bias.

4 STUDY AREA AND DATASETS USED

This chapter introduces the study area in the first part and provides the background information behind the motivation of conducting the study in this area. The second part of this chapter is focused on explaining the datasets that have been considered for the implementation of the designed algorithms.

4.1 Study Area

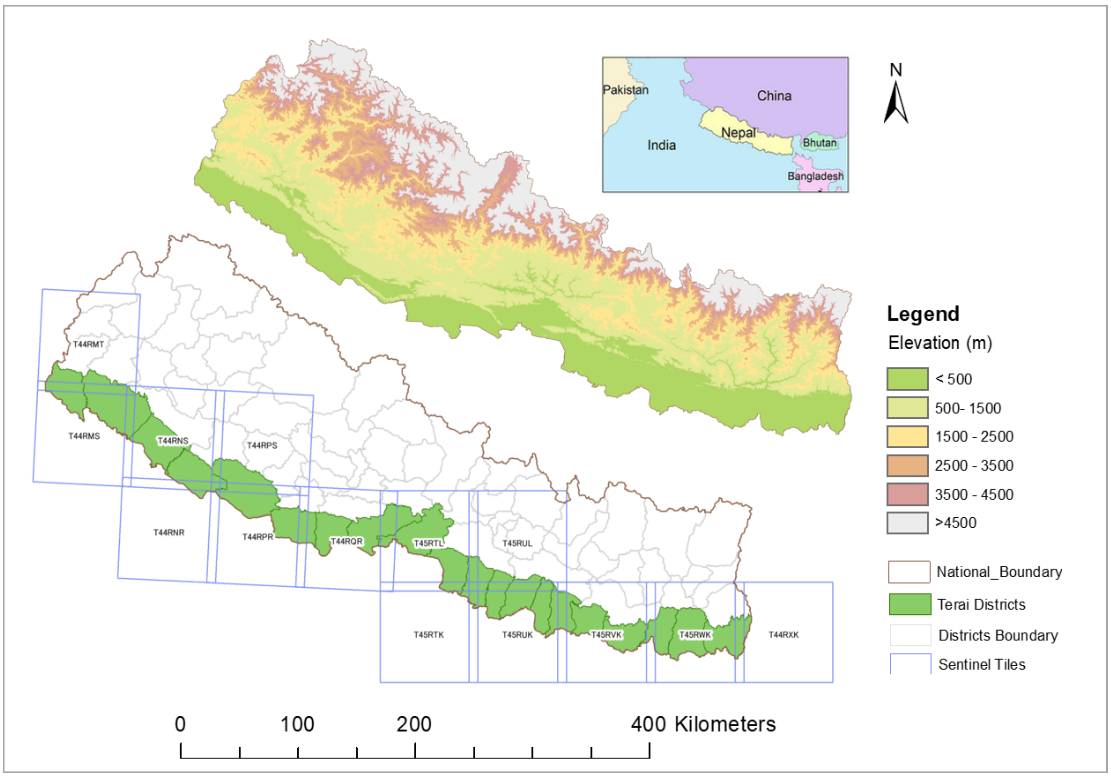

Figure 4.1: Study area with elevation profile and sentinel tiles

This study is based on 20 districts of the Terai region of Nepal that comprise a lowland in southern Nepal (Figure 4.1). Nepal is a small landlocked country with an area of 147,181 km^2 stretching from 26° 22′ to 30° 27′ N and from 80° 04′ to 88° 12′ E. The principal economic activity of Nepal is agriculture, which constitutes about one-third of the Gross Domestic Product (GDP) and employs nearly three-fourth of the labor force [57]. Of the total basic crop production, paddy production is the highest in the country sharing 20.75% of the total GDP from agriculture [58]. Even though the Terai region covers just 23.1% of the total area of the country, it comprises 49% of the total agricultural land. The rice production is mainly located on the Terai districts which contributes about 70% of the total rice production of the country. As a result, the domestic food security of Nepal is critically reliant on the sustainability of the cereal production system of this region. According to CBS (2011), 84% of the farm households in the Terai region are actively engaged in rice production. However, the growth rate of the agricultural sector of Nepal is too low to meet the growing food demand of the increasing population, indicating that public and private investments in the agricultural Research and Development (R&D) sector would increase cereal productivity in Nepal [59].

4.2 Datasets used

4.2.1 S2 Level-1C products

S2 Bands	Band Names	Central Wavelength (nm)	Bandwidth (nm)	Spatial Resolution (m)
Band 1	Coastal Aerosol	443	20	60
Band 2	Blue	490	65	10
Band 3	Green	560	35	10
Band 4	Red	665	30	10
Band 5	Vegetation Red Edge	705	15	20
Band 6	Vegetation Red Edge	740	15	20
Band 7	Vegetation Red Edge	783	20	20
Band 8a	Near-Infrared	842	115	10
Band 8b	Vegetation Red Edge	865	20	20
Band 9	Water Vapour	945	20	60
Band 10	SWIR-Cirrus	1375	30	60
Band 11	SWIR	1610	90	20
Band12	SWIR	2190	180	20

Table 4.1: S2 spectral bands definition, Source: [60]

The key phases of the rice crop cycle, i.e. start of the season, the peak of season and end of the season, in the Terai region of Nepal correspond to mid-July, mid-September, and mid-November respectively [10]. Therefore, the L1C products during these key periods were downloaded for the years 2016-2018. In case of missing or corrupted images, products within a week before or after the mid-month were downloaded. 14 sentinel tiles (zone

44 and 45) in total covered the study area as shown in (Figure 4.1. Therefore, a total of 126 L1C products were downloaded. The S2 L1C products are provided free from ESA's Scientific Data Hub (SDH) in Standard Archive Format for Europe Format (SAFE) files. S2 sensor provides a total of 13 spectral bands with a high spatial resolution ranging from 10m to 60m (Table 4.1). Among these spectral bands, the classical RGB and Near Infrared (NIR) bands with 10m spatial resolution are dedicated to land applications. The 20m bands which 4 narrow bands in the vegetation red edge spectral domain and 2 Shortwave Infrared (SWIR) large bands, which are used for snow/ice/cloud detection and moisture stress assessment. The remaining bands at 60m are dedicated to atmospheric correction and cirrus detection.

Both satellites have a wide-swath of 290 km and fly in the same orbit phased at 180° providing a high temporal resolution with a revisit frequency of 5 days for S2 operational data products [61]. S2 L1C and Level-2A (L2A) products are provided in tiles, also called granules, which consist of 100× 100 km^2 ortho-images in UTM/WGS84 projection. The Universal Transverse Mercator (UTM)system divides the Earth's surface into 60 zones. Each UTM has a vertical width of 6° of longitude and a horizontal width of 8° of latitude. On the one hand, L1C products are top-of-atmosphere Top-of-Atmosphere (TOA) radiance images which are radiometrically and geometrically corrected as well as orthorectified using the global Digital Elevation Model (DEM). On the other hand, L2A products provide Bottom-of-Atmosphere (BOA) reflectance images derived from the corresponding L1C products. For this study, L1C products are considered since L2A products are only available from the end of 2018 for regions outside Europe.

Sentinel Data Processing

A total of 126 S2 L1C products were converted to atmospherically corrected L2A products using the Sen2Cor processor which is based on algorithms proposed in the Atmospheric/-Topographic Correction for Satellite Imagery (ATCOR) [62]. Sen2Cor is supported by ESA as a third-party plugin for the S2 toolbox (standalone version). It runs in the ESA Sentinel Application Platform (SNAP) or from the command line. Additionally, topographic correction with a 90m digital elevation database from CGIAR-CSI (http://www.cgiar-csi.org) and cirrus corrections were applied [63]. Topographic correction here is purely radiometric and does not change the image geometry. However, the resulting products showed that cirrus correction with Sen2Cor was not effective. Moreover, the products had significant cloud and cloud shadows. So, to deal with these problems, cloud mask data available with L1C products that specifies the percentage of cloudy pixels and cirrus pixels, was considered during the post processing of L2A products. The whole process was performed in batch processing from the command prompt. The whole process was performed in batch processing from the command prompt.

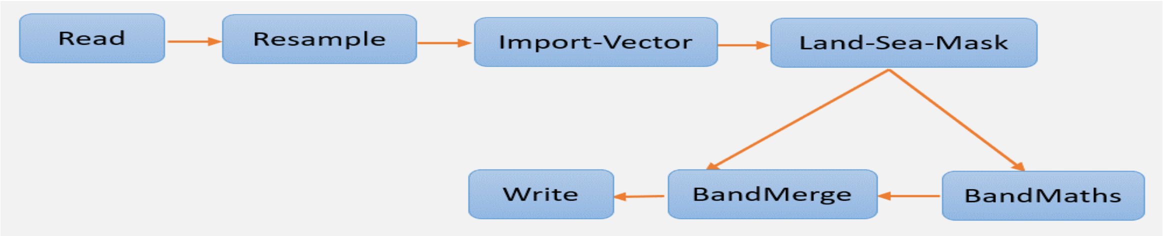

Figure 4.2: S2 L2A data processing chain in SNAP Graph Builder

After generating L2C products, the next step was to calculate the NDVI. This process involves a chain of data processing tools as shown in Figure 4.2. Taking into account the total size of the considered products (over 100 GB) and the high computational cost of managing such data volume, we decided to store the final products in 20 m instead of 10 m. Therefore, the resulting L2A products were all resampled to 20 m. Thereafter, a spatial subset was applied to clip the images to the extent of the study area. Among the 13 spectral bands of S2, the three bands with 60 m resolution (B01, B09, and B10) are dedicated to atmospheric correction and cirrus detection. These bands are typically not considered in crop classification tasks [7] so, they were excluded from the output data. As a result, 4 bands resampled to 20 m (B02-B04 and B8a) and 6 bands with a nominal spatial resolution of 20 m (B05-B07, B8b, B11, and B12) were concatenated for the considered data processing chain. Bands B04 and B8a bands represent the red and NIR channels respectively (see Table 4.1) and were used to calculate the NDVI using BandMaths tool as follows:

$$NDVI = \frac{(\rho_{nir} - \rho_{red})}{(\rho_{nir} + \rho_{red})}, \tag{4.1}$$

where ρ_{nir} represents the reflectance of the near-infrared band and ρ_{red} represents the reflectance of the red band. NDVI is the most commonly used indicator to monitor vegetation health and classifies vegetation extent. As a result, the resulting NDVI product was also merged as an additional band to the final product in the GeoTIFF file format. Considering the presence of significant cloud coverage in the images, cloud mask data available with L1C products was also concatenated with the bands. An option is to use these cloud masks to filter the images when training the corresponding crop classification models. Due to the high cloud cover in June (heavy monsoon period), only the images of September and November have been considered for the classification and yield estimation experiments reported in this work.

4.2.2 Ground-Truth Data: Rice Mask

Considering the major limitation on the availability of a real high-resolution ground-truth map of rice crops in the study area, the rice mapping approach adopted by [10] was utilized in the long-term period 2006-2014 to generate the final ground-truth rice map considered in this work. Specifically, this procedure considers the most optimistic rice map based on the classification conducted using MODIS data. The resulting rice map has also been filtered by using the land cover mask produced by [22] in order to ensure that the ground-truth maps do not include non-agricultural areas if any. Finally, the output

S.N.	Climate Variables	Unit
1.	Rainfall	Millimeter (mm)
2.	Maximum temperature	Degree Celsius ($^{\circ}C$)
3.	Minimum temperature	Degree Celsius ($^{\circ}C$)
4.	Relative Humidity	Percentage (%)

Table 4.2: Climate data used for yield estimation purpose

map was re-projected to the UTM/ WGS84 projection and resampled into 20 m to extract the corresponding ground-truth rice labels for the S2 spatial resolution.

4.2.3 Rice Crop Yield data

We set rice yield data published by the Ministry of Agriculture and Livestock Development (https://mold.gov.np/), Government of Nepal, as a target of estimation. The yield data was downloaded for the considered years 2016-2018. While most of the studies discussed in section 2.2.2 were conducted using county-level data, we only have the option of using district-level data (larger administrative unit). The scarcity in ground truth yield data is a major challenge in developing countries like Nepal. To feed the yield data as ground-truth labels in the network, firstly, the yield values in kilogram per hectare (kg/ha) is converted to kg/pixel where an area of each pixel is 400 square meter. Secondly, the rice pixels from rice mask used in classification tasks are labeled with these values. The final results are then again summarized to kg/ha.

4.2.4 Auxiliary data

The agricultural practice in the study area is mostly dependent on natural irrigation due to lack of irrigation facilities which is one of the production constraints in the study area. As climatic variables have a significant impact on crop production, they are also used as input datasets in addition to S2 data, for yield estimation process [25]. Furthermore, the authors in [42] suggested that in addition to spectral data, soil and climate data can contribute to further improvements to crop yield estimation results. The auxiliary data that are used in this study for rice yield estimation process are explained below:

Climate data

The climatic data that are used in this study are listed in Table 4.2. These data are made available from the Department of Hydrology and Meteorology, Government of Nepal (https://www.dhm.gov.np/). Considering the time period during which the images were downloaded, 15 days average (1 week before and after the considered date) of each of these climatic variables was calculated from the available daily data. Spatial interpolation of these data was performed using the ordinary kriging method as supported by [64][65] and the dataset of the whole country was used for this. The experimental space-time semivariogram was calculated for each climate data and for each time period.

Leave one out cross-validation method was used to assess the error associated with the model with parameters, producing a Mean Error (ME)and RMSE. The model parameter with least ME and RMSE are used for surface generation of particular climate data. The resulting raster data were then normalized within the range of [0-1] by using min-max normalization, which is a general procedure used in machine learning algorithms. The min-max normalization is done using the following formula:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}\ , \tag{4.2}$$

where x denotes the original value and x_{norm} is the resulting normalized value. After normalization, similar to the ground-truth rice mask, the normalized climate data in raster format were re-projected, re-sampled, clipped to tiles maintaining the spatial extent of tiles.

Soil Data

Soil data of spatial resolution 250m were downloaded from Krishi Prabidhi Project site (https://krishiprabidhi.net/). The Krishi Prabidhi project is led by the International Maize and Wheat Improvement Centre and collaborates with different divisions of MOALD and private sector for increasing national crop productivity, economic welfare, and household-level food security. The soil data includes six variables namely:

- Boron Contain,
- Clay Contain,
- Organic Matter,
- PH,
- Sand, and
- Total Nitrogen.

The soil data were also normalized within the range of [0-1]. Like previous data, these data were also re-projected, re-sampled to 20m, clipped to tiles maintaining the spatial extent of tiles and finally stacked as a single layer with multiple bands in order to be used in the CNN model

5 METHODOLOGICAL DESCRIPTION

This chapter explains the methodology adopted for the implementation of chosen network architectures and the approach used for their performance evaluation. The first half starts with an explanation of network architectures of all the classification models. This is followed by deep insights on the procedure for network training, experimental settings, classification, and performance evaluation based on qualitative and quantitative approaches. Similarly, the second half is concerned with the details of network architecture used for yield estimation purpose together with the detailed process used for network training, experimental setup, yield estimation and evaluation of models' performance.

5.1 Classification

5.1.1 Classification Models

For rice crop classification purpose, firstly, we are considering a simple 2D CNN architecture of the ConvNet network which will be referred to as ConvNet 2D-1 hereafter. Additionally, we are considering the architectures that we have chosen based on a review of related works explained in sections 2.1.2 and 2.1.3. The second 2D CNN architecture based on [27], will be denoted as ConvNet 2D-2. Similarly, the third form of 2D CNN and the fourth architecture, 3D CNN based on [33] will be mentioned as ConvNet 2D-3 and ConvNet 3D respectively henceforward. The three variations of 2D CNN have been employed to analyze how the classification accuracy varies with the change in network architecture. Furthermore, SVM is employed as a baseline algorithm.

5.1.1.1 Patch-Based CNN Network Architectures

CNNs extend the classical artificial neural network approach by adding multiple convolutional layers and filters that allow representing the input data in a hierarchical way [50]. All the CNN architectures chosen for crop classification purpose follow a typical CNN architecture with a concatenation of convolution and pooling layers followed by the fully connected layer(s). Additionally, all four architectures are patch-based CNN which extract image patches and classify the center pixel of the patch. A patch-based CNN is proven to perform better than pixel-based CNN [27][66]. It considers the spatial relationship of a pixel to its neighborhood and the salient features in a patch are indicative of the belonging to one class or another. The following subsections give a brief explanation of each architecture:

ConvNet 2D-1

For a 2D CNN network architecture, all the spectral bands of the available time period are stacked up to form a 3D tensor. The first 2D CNN model, ConvNet 2D-1 comprises of 4-layer as shown in Figure 5.1. The initial three layers consist of the concatenation of convolution and pooling with 8, 16 and 32 number of filters respectively. The number of filters is doubled with each subsequent convolutional layer to increase the number of feature maps in the hidden layers. A small kernel size of 3 × 3 is used to represent the multi-level features. The rectified nonlinear activation function (ReLU) is performed after every convolution to introduce non-linearity to the CNN and is followed by the batch normalization layer. Then a max-pooling layer follows with size 2 × 2 and stride 2 that down samples the spatial dimension of the input and reduces the computational burden. In the last layer i.e., the FC layer, each neuron provides a full connection to all the learned feature maps issued from the previous layers. The fully connected layer together with a softmax activation at the end uses learned high-level features to classify the input images into predefined classes: rice and non-rice.

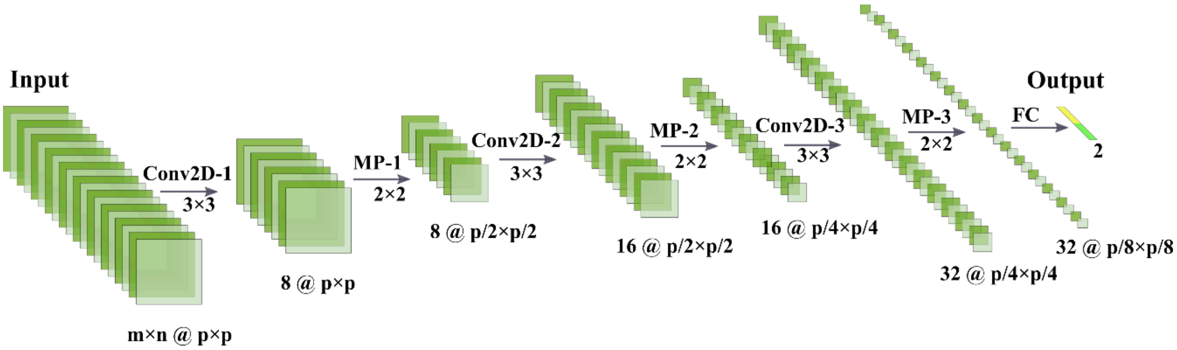

Figure 5.1: Architecture of ConvNet 2D-1 (Conv2D: 2D convolution, MP: Max-pooling, FC: Fully connected, m: number of channels, n: temporal depth, p: patch size)

ConvNet 2D-2

The second 2D CNN model, ConvNet 2D-2 (Figure 5.2 is also a 4-layered network but differs from the first in terms of the number of convolutional layers, kernel size and numbers and number of fully connected layers. This network architecture consists of 2 convolutional layers with 400 and 800 number of filters of size 5 × 5, which is much higher than the ConvNet 2D-1. Two fully connected layers with 1000 and 2 neurons respectively follow at the output end. The first fully connected layer collects all the output from the previous layers as a flat array and finally, the last fully connected layer classifies into rice and non-rice as like in the previous model.

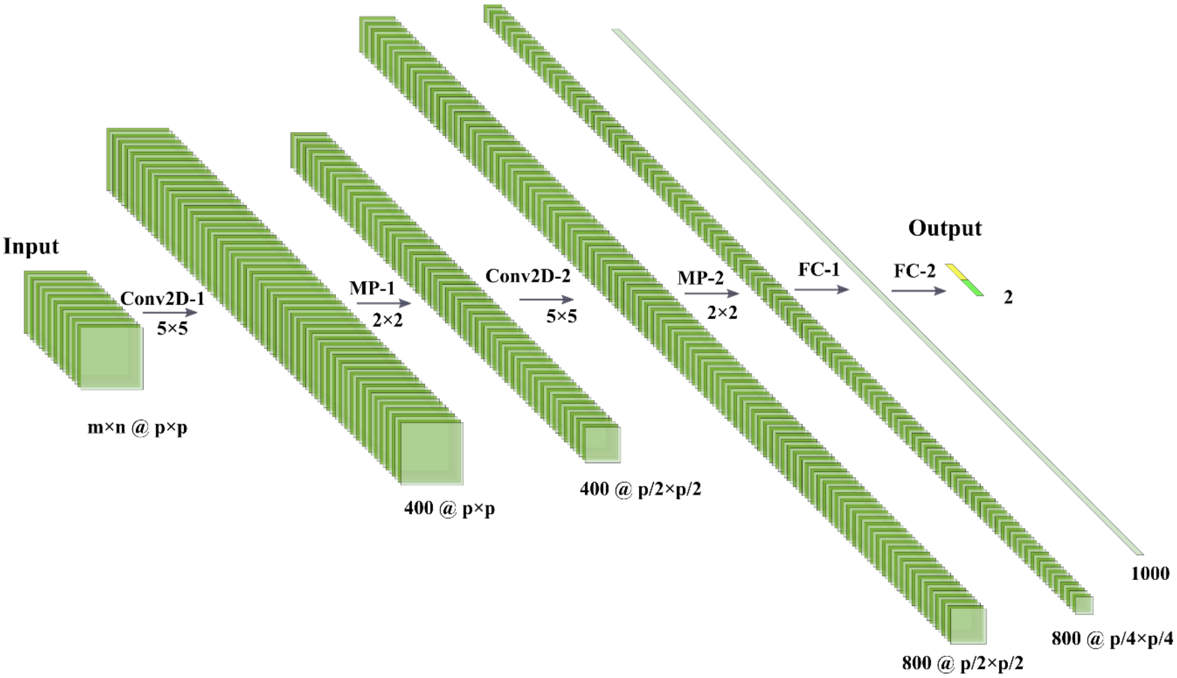

Figure 5.2: Architecture of ConvNet 2D-2 (Conv2D: 2D convolution, MP: Max-pooling, FC: Fully connected, m: number of channels, n: temporal depth, p: patch size)

ConvNet 2D-3

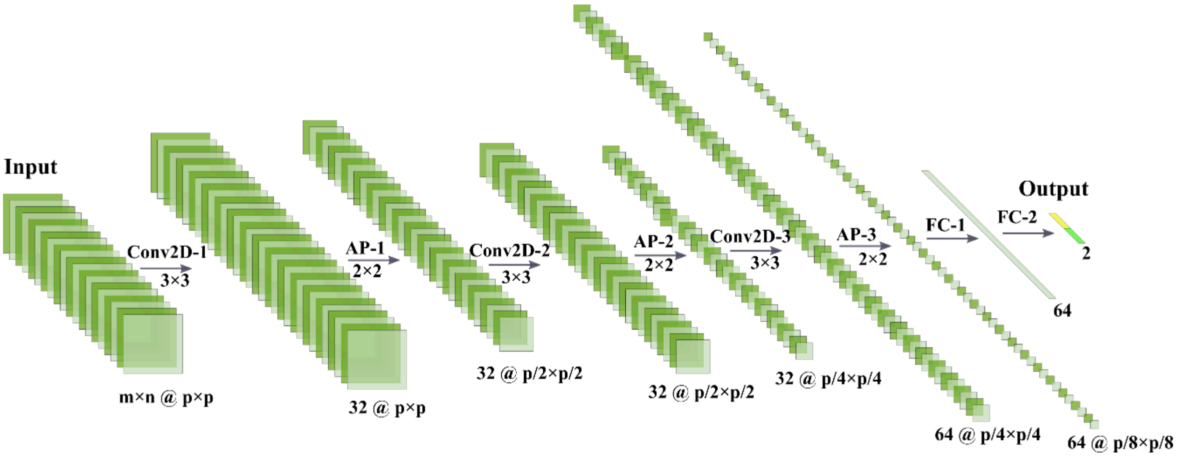

Figure 5.3: Architecture of ConvNet 2D-3 (Conv2D: 2D convolution, AP: Average-pooling, FC: Fully connected, m: number of channels, n: temporal depth, p: patch size)

The final form of 2D CNN, named ConvNet 2D-3 (Figure 5.3) has taken a neural network developed by Oxford's Visual Geometry Group (VGGnet) [32] as a template to train the CNN. Unlike previous two 2D CNN models, this network architecture comprises 5 layers among which the first three are convolutional layers with 32, 32 and 64 number of kernels of size 3 × 3. Another difference in this model is the use of average pooling instead of max-pooling after each convolution. The last two layers are fully connected layers with 64 and 2 neurons respectively.

ConvNet 3D

The final model considered for classification experiments is ConvNet 3D (Figure 5.4) which is exactly the same with ConvNet 2D-3 except for the way input images are fed to the network and all 2D convolution operations are replaced by 3D. Unlike in 2D CNN, in 3D CNN the temporal information is stored in a separated dimension which results in 4D tensor (spatial, spectral and temporal dimensions). Since temporal information is available for just two time periods for each year, the temporal dimension is preserved until the first fully connected layer by using a pooling layer of size 2 × 2 × 1.

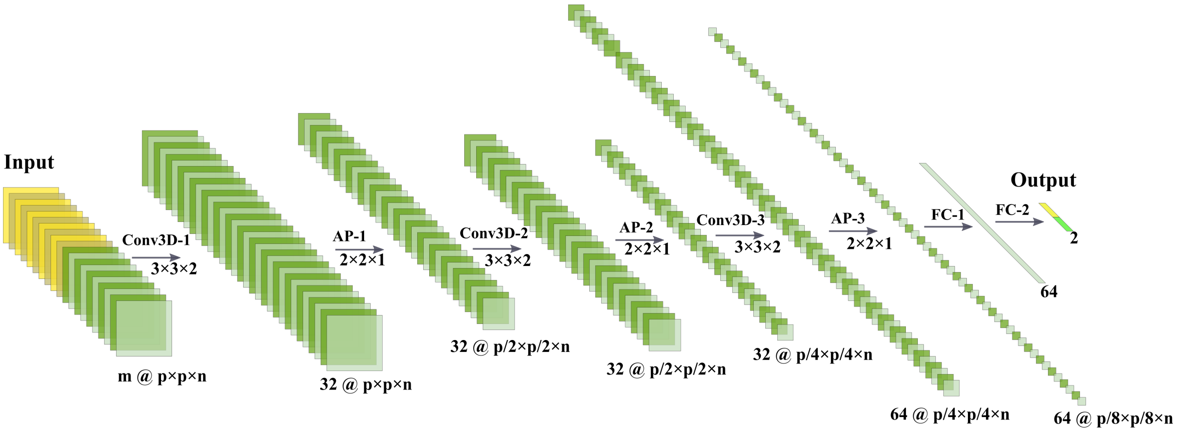

Figure 5.4: Architecture of ConvNet 3D (Conv3D: 3D convolution, AP: Average-pooling, FC: Fully connected, m: number of channels, n: temporal depth, p: patch size)

5.1.1.2 Baseline Algorithm: SVM

The initial experiment in this study with the RBF kernel revealed that the computational cost in terms of time is not affordable for such large datasets. As a linear SVM has been tested and proposed as optimal when the number of features is large [29], we also employed a linear SVM with pixel-based inputs.

5.1.2 Network Training

In order to maintain the same environment of data inputs, we considered only the S2 spectral images as input in all the models. As S2 images are processed in tile levels (5490 × 5490 pixels), the ground truth rice mask is also exported in tiles. All the network architecture explained in the previous section are patch-based CNN architecture. Therefore, we used image patches centered in the pixel of interest. Note that only those patches with the central pixel belonging to the region of Terai, also termed as valid patches, were extracted. In all cases, the input patch is abstracted into multi-level representations to classify the central pixel.

Before extracting image patches, firstly, the NDVI threshold (below 0.1) was applied to mask out the non-vegetation areas from Sentinel bands. This process also the clouds since they are characterized by negative NDVI values. Therefore, the cloud masks from S2 L1C products which were considered as an option to filter the clouds from the images were not essential after applying the NDVI threshold. Secondly, the most optimal ground truth pixels were generated considering the availability of the corresponding S2 pixels in all the temporal images. On the one hand, in case of 2D CNNs, all the spectral bands of n temporal dimension (2 in this case) are stacked together to feed the network while on the other hand, in case of 3D CNN, temporal feature are stored separately in another dimension so that the temporal information does not collapse in the first convolution operation.

The general procedure of the training stage is shown in Figure 5.7. Image patches with corresponding GT labels are input to the classification models as the training samples. The *Softmax* function is then performed on the output feature map generated by the last fully connected layer to predict the class distribution in categorical output. Thereafter, a cross-entropy loss is calculated by comparing the predicted labels against GT labels. This loss value is back propagated to update the network parameters using SGD with momentum.

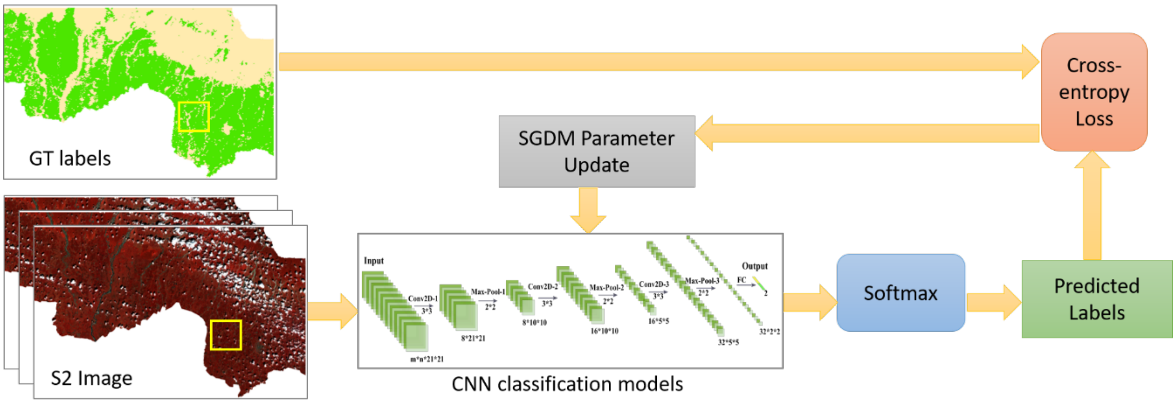

Figure 5.5: General procedure of training classification networks (modified after [30])

5.1.3 Experimental Settings

The experiments carried out in this study (classification and yield estimation) were built on the top of deep learning framework of MATLAB 2019b and were executed on a server with Intel(R) Core (TM) i7-6850K processor with 64 Gb RAM and 2 GPUs (GeForce RTX 2080 Ti 11GB and GeForce GTX 1080 Ti 11GB) for parallel processing.

The input data consists of m × n patches with p × p size where m and n are the numbers in the spectral and temporal dimension respectively, and p is the patch size with the same width and height. In our case, the number of spectral bands is 10 and the temporal dimension is 2. In the initial experiments, patch size (p) is set to 21 × 21 and is fine-tuned later along with the other hyperparameters. To validate and optimize the classification models, all the experiments are conducted with data from 2016. To ensure maximum datasets for training, from the 14 tiles covering the study area (Figure 4.1), 13 tiles were used for training the network (considering a 10% of this data for validation purposes) and the remaining tile for the test. While an ideal case would be training and testing the networks with each of the 14 tiles out, one at a time and averaging the classification result, it is important to note that computational time required for training process is expensive and unaffordable with this scenario. Therefore, we used this one-tile-out scheme for only three random test tiles (T45RVK, T45RUK, and T44RQR) in order to generate an average classification result for all the tested models.

During the experiments, the training set was used to train the network while the validation dataset was used to fine-tune hyperparameters and to perform early stopping. Overall accuracy was considered as a criterion for evaluating these experiments. All the networks are trained using the SGD with momentum for a maximum of 30 epochs (Table 5.1). The training data is shuffled before each training epoch and the validation data is shuffled before each network validation. Early stopping is integrated into the training process to stop the training after *n* non-improving iterations to avoid the overfitting of the model. The hyperparameter *n* is called patience [67] which is set to 100 in all the experiments. Momentum and L2 regularization value were set to 0.9 and 0.00005 [27].

Hyper-parameters	**Optimizer**	**Epochs**	**Validation Patience**	**Momentum**	**L2 Regular-ization**
Values	SGD with momentum	30 (Max)	100	0.9	0.00005

Table 5.1: Hyperparameters values for CNN models

5.1.3.1 Sensitivity to Hyperparameters

The hyperparameters that were investigated in classification experiments were patch size, learning rate, and mini-batch size (Table 5.1). The values of these hyperparameters to be tested are selected based on the reference network architectures from [27] and [33]. Note that we are maintaining a general scheme to compare the performance of models in fair conditions. During the experiments, when one hyperparameter's value is changed, other hyperparameters are kept constant to determine the optimal hyperparameter values.

Hyperparameters	**Value**
Patch size	9,15,21
Learning rate	0.01, 0.001
Mini-batch size	100, 500

Table 5.2: CNN sensitivity experiments on hyperparameters (Patch size, Learning rate, and Mini-batch size)

Patch size:

In a patch-based CNN, the center pixel of the patch is classified considering its spatial relationship with the neighboring pixels and the salient features of the patch. It is therefore important to determine an appropriate patch size that can capture spatially, local correlation of the center pixel to the surrounding pixels. To evaluate the performance of

CNN architecture with the varying patch sizes, 3 patch sizes; 9, 15 and 21 were selected (Table 5.2). Considering the presence of a significant amount of noise (cloud and shadows) which could cause mixed pixels effect [33], patch size higher than 21 was not considered.

Learning Rate:

Learning rate is a hyperparameter that tests how much to change the model each time the weights of the model are updated in response to the predicted error. It is one of the most important hyperparameters to be tuned while configuring a neural network. The effect of the learning rate was investigated by varying its values during the experiments while maintaining a fixed configuration of other parameters and hyperparameters. Two values of learning rate: 0.01 and 0.001 were tested during the experiments. These values are chosen from the papers from which the network architectures are adapted.

Mini-batch Size:

The optimization that uses mini-batch gradient descent divides the training data into small batches that are used to calculate error in the model and update model coefficients. Two values of mini-batch: 100 and 500 were tested in the experiments.

5.1.3.2 Significance of Multi-temporal Inputs

As mentioned before in section 4.2.1, the images of June (start of the season) could not be used due to high cloud cover and therefore, images of only two time periods have been used for all the experiments. In this scenario, one more experiment was conducted to verify if temporal information from images of only two time periods (September and November) is adding contribution to the performance of the classifiers in comparison to the use of single time period (September) images.

5.1.4 Performance Evaluation

During the experiments to select the optimal hyperparameters, the performance of the models was evaluated using overall accuracy metrics. After these experiments, the models with the best hyperparameter combinations were selected for final training and classification of the input dataset. The final classification results were evaluated quantitatively using metrics namely overall accuracy and F1-score statistics and qualitatively through visual inspection of maps. The details about these measures are presented below:

5.1.4.1 Quantitative Approach

In this process, the classifiers were evaluated with the test dataset against the ground truth using these performance metrics. The metrics are based on four classification outputs as shown in the confusion matrix in Table 5.3. This is because we are considering rice mapping as a binary classification where rice pixels are positive class and non-rice pixels are negative.

Actual Class		**Predicted Class/ Classified Class**	
		True	**False**
	True	True Positive	False Negative
	False	False Positive	True Negative

Table 5.3: Confusion matrix (Description of TP, FP, TN, and FN)

In a binary classification model, each instance/pixel is classified into two classes, true and false classes. This gives rise to four possible classifications for each instance which are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) [25]. TP indicates the number of correct predictions that a target pixel (rice pixel) is positive (rice class); FP denotes the number of non-target pixels (non-rice pixels) classified as positive; TN is the number of correctly classified non-target pixels, and FN denotes the number falsely classified target pixels as non-targets. In Table 5.3, the classifications that are shown on the diagonal are the correct classifications which are TP and TN. From the confusion matrix, a number of performance metrics can be derived.

Precision and Recall

Precision refers to the ratio of correctly classified positive pixels to all the predicted positive pixels. In other words, precision is an indication of out of predicted positive, how many of them are positive. Recall refers to that of correctly classified positive pixels to all observations in the actual class. Precision is useful when the cost of FP is high while recall is important when the cost of FN is high. Precision and recall can be computed as [25]:

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{5.1}$$

Overall Accuracy

Overall accuracy is the measure of all the correctly identified classes. It is mostly used when all classes are equally important i.e. when TP and TN are more important. Overall accuracy is usually expressed as a percentage, with 100% accuracy being the corrected classification where all inputs are correctly classified.

$$Overall Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.2}$$

F1-Score

F1-score is the harmonic mean of precision and recall. It takes both false positive and false negative into account. These metrics might be a better measure when there is an uneven class distribution (a large number of true negatives). F1- score is calculated as:

$$F1 = \frac{2 \times Precision \times Recall}{Precesion \times Recall} \tag{5.3}$$

5.1.4.2 Qualitative Approach

Once the hyperparameters are tuned, the trained models with the best hyperparameters combination are used to classify the images into maps representing rice and non-rice. During this process, patch-based multitemporal images are defined as input along with the selected training model. The image patches are then classified into predicted labels corresponding to the predefined classes in an iterative manner to generate the classified map.

5.2 Yield Estimation

To address the second part of the study which is rice yield estimation, firstly we considered a simple 2D CNN architecture of the ConvNet network. Additionally, we implemented the chosen 3D CNN network architecture as discussed in section 2.2.3. As a baseline algorithm, SVR was employed.

5.2.1 Network Architecture

Similar to the CNN models used for rice classification, both the 2D CNN and 3D CNN models that are implemented for estimating the rice yield are patch-based i.e. for each image patch extracted, rice yield for the center pixel of the patch is estimated. The following subsections briefly describe the architecture of these models:

2D CNN

The network architecture of 2D CNN for rice yield estimation is exactly similar to the ConvNet 2D-1 network (Figure 5.1) which was designed for rice classification purpose, except that the classification layer is replaced by the regression layer. While the last fully connected layer in the classification models has two neurons for two predefined classes: rice and non-rice, the FC layer for the regression task consists of 1 neuron to estimate rice yield.

3D CNN

The author in [13] used a channel compression module to reduce the channel dimension from 10 to 3 without altering the spatial or temporal dimensions considering that all the 10 channels are not of equal importance. Taking this into account, we also started our network using the dimensionality reduction technique to reduce the number of channels to 3. For this, the 3D CNN network architecture consists of two 3D-convolutional layers at the beginning. The first layer takes the input tensor and uses 10 filters performing 3D-convolutions with kernel size $3 \times 3 \times 2$. The second layer takes the output from the first layer and performs 3D-convolution with kernel size $1 \times 1 \times 2$ using 3 filters. Figure 5.6 shows the architecture of 3D CNN and Table 5.4 lists the details of each layer of the architecture.

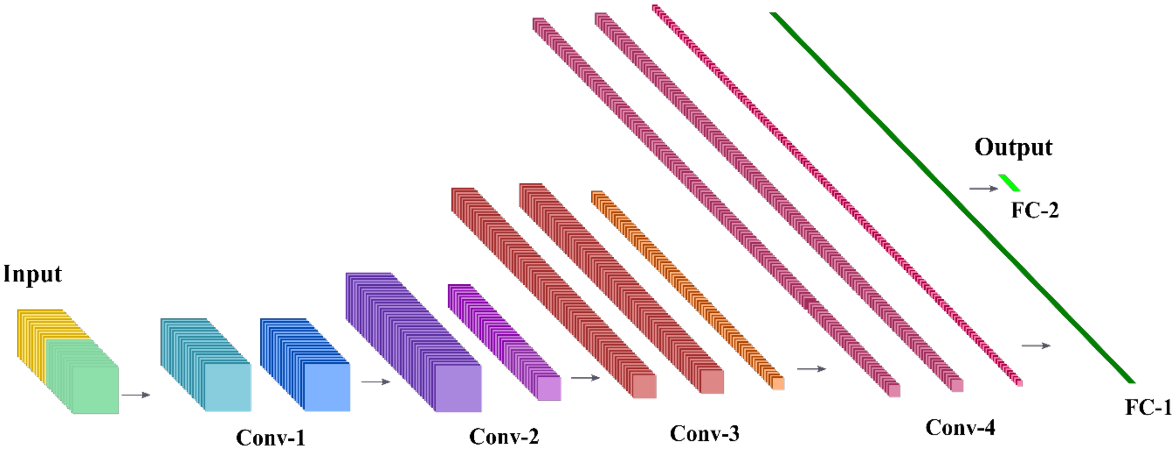

Figure 5.6: Architecture of 3D CNN. The detail architecture of each layer is listed in Table 5.4

.

Blocks	Layers	Filters	Kernel Size	Stride	Padding	Output Size
DR	DR-1 (conv3D)	10	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	10 × 2 × 21 × 21
	DR-2 (conv3D)	3	1 × 1 × 2	1 × 1 × 1	0 × 0 × 1	3 × 2 × 21 × 21
Conv-1	Conv3D-1	64	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	64 × 2 × 21 × 21
	Max-Pool-1	-	2 × 2 × 1	1 × 1 × 1	-	64 × 2 × 21 × 21
Conv-2	Conv3D-2	128	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	128 × 2 × 21 × 21
	Max-Pool-2	-	2 × 2 × 1	2 × 2 × 1	-	128 × 2 × 10 × 10
Conv-3	Conv3D-3a	256	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	256 × 2 × 10 × 10
	Conv3D-3b	256	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	256 × 2 × 10 × 10
	Max-Pool-3	-	2 × 2 × 1	2 × 2 × 1	-	256 × 2 × 5 × 5
Conv-4	Conv3D-4a	512	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	512 × 2 × 5 × 5
	Conv3D-4b	512	3 × 3 × 2	1 × 1 × 1	1 × 1 × 1	512 × 2 × 5 × 5
	Max-Pool-4	-	2 × 2 × 2	2 × 2 × 2	-	512 × 1 × 2 × 2
FC-1	FC-1	-	-	-	-	1024
FC-2	FC-2	-	-	-	-	1

Table 5.4: Layers architecture of the 3D CNN

After the dimensionality reduction (DR) module, the network is composed of the concatenation of convolutional blocks and max-pooling layers. The convolutional blocks are composed of 3D convolutional layers with 3 × 3 × 2 kernel, stride 1 × 1 × 1 and padding 1 × 1 × 1 which are followed by batch normalization layer and ReLU activation layer. The convolutional blocks are then followed by max-pooling layers with kernel and stride 2 × 2 × 1 to maintain the temporal depth. The spatial dimension of the input data is maintained the same until the second convolutional layers after the channel compression module which is then reduced to half after each max-pooling layer. Considering the limitation in the temporal dimension, temporal depth is maintained until the last convolutional layer and then reduced to half by the last max-pooling layer. The output features from the last max-pooling layer are then flattened into a fully connected layer with 1024 neurons which is followed by the final FC layer with 1 neuron for predicted yield values. The regression layer at the end outputs the regressed values and calculates layer computes the half-mean-squared-loss during the model training process.

5.2.2 Network Training

For the yield estimation task, the number of input channels for each timestamp is 20 (10 S2 bands, 4 climate bands, and 6 soil bands). Each channel is normalized to the range of [0,1] using the min-max normalization technique before feeding the input data to the network. NDVI threshold below 0.1 is applied in these models as well, to filter non-vegetation and cloud pixels from the input data. For ground truth labels, per pixel yield (kg/pixel) for each district is calculated from district level production data. Note that only those patches were considered valid, which do not contain no-data pixels and whose center pixel contains yield information to avoid noise during the training process.

The general procedure of the training stage in the estimation process is shown in Figure 5.7. The valid input patches with corresponding GT labels are input to the yield estimation models as the training samples. In the network architecture, convolution and pooling layers extract important features from the input images and the level of extraction/ abstraction depends upon the complexity of the network. For regression problems, a fully connected layer typically precedes the regression layer. In 3D CNN, two FC layers are used where the first FC layer in network collects all the output from the previous layers as a flat array and the last fully connected layer predicts the yield corresponding to the input sequence. The regression layer predicts the regression output and computes the half-mean-squared-loss by comparing it against the GT labels. This loss is backpropagated to update the network parameters using Adam optimization.

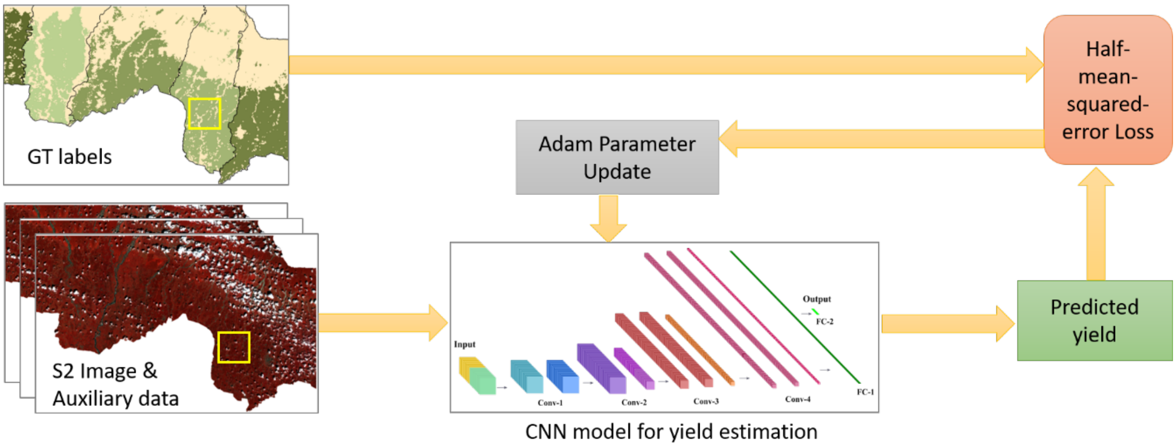

Figure 5.7: General procedure of training yield estimation networks (modified after [31])

5.2.3 Experimental Settings

The input for the CNN models for yield estimation is similar to that for classification i.e. it consists of m × n patches with p × p size where m and n are the numbers of channels and temporal depth respectively, and p is the patch size with same width and height. Patch size (p) is set to 21 × 21 as it is proven to be the optimal patch size in the classification experiments. The number of channels (m) is 20 and temporal depth (n) is 2. The networks are trained using Adam optimizer for 30 epochs. The learning rate and mini-batch size are set to 0.001 and 100 respectively [13]. To train, validate and test the performance of the CNN models, the data for all the available years (2016-2018) are used. Early stopping is integrated into the model based on the validation dataset in order to avoid the overfitting of the model. We have designed two experimental setups based on how the whole dataset is divided into training and validation/test datasets. RMSE in kg/ha is used as a criterion to evaluate the results of experiments.

Experiment 1:

In the first experiment, from the whole dataset of three years, 70% is used for training and the remaining 30% for validating the models.

Experiment 2:

In the second experiment, data from 11 out of 14 tiles were used to train the models while the remaining 3 tiles (T45RVK, T45RWK, and T45RXK) which cover 5 districts, were used to test the performance of the model.

Under these experimental setups, we will be testing the following scenarios:

1. **Variation in ground truth data**

   Taking into account that the ground truth data is very sparse (district level), two approaches are considered to feed the ground truth labels to train the network.

   a) The first approach is to calculate the yield per pixel (kg/pixel) by dividing the total production of each district by the total number of rice pixels in that district so that each rice pixel of a particular district is labeled with the same yield value.

   b) The second approach is to vary the per pixel yield (by 10%) in proportion to corresponding NDVI values of pixels in the peak of the season (September). NDVI based yield variation is based on the assumption that the higher the NDVI values, the healthier the vegetation and the larger the yield and vice versa.

   Let $Y_o(i,j)$ be original rice yield value of a pixel at (i,j) obtained from the first approach a), and $Y_o'(i,j)$ be new yield value at corresponding pixel after variation. The average NDVI value n_a over the image of dimension $M \times N$ can be calculated as

$$n_a = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{n_{(i,j)}}{MN}, \tag{5.4}$$

where $n_{(i,j)}$ be NDVI value at pixel (i, j). The variation factor for NDVI value at each pixel (i, j), $f_{(i,j)}$is calculated as:

$$f_{(i,j)} = \frac{n_{(i,j)}}{n_a} \tag{5.5}$$

Now, the original yield value $Y_o(i, j)$ can be transformed into new yield value $Y_o'(i, j)$ using this variation factor with variation percentage v of 10% as follows:

$$Y_o'(i,j) = \begin{cases} Y_o(i,j) + vY_o(i,j)f_{(i,j)}, & if\ f_{(i,j)} > 1 \\ Y_o(i,j), & if\ f_{(i,j)} = 1 \\ Y_o(i,j) - vY_o(i,j)f_{(i,j)}, & if\ f_{(i,j)} < 1 \end{cases}$$

2. **Testing the significance of channel compression module in 3DCNN**

   As suggested by [13] for future work, we also tested whether the channel compression module contributes to the performance of the 3D CNN model besides reducing the computational cost. To perform this test, we will evaluate the performance of the 3D CNN model with and without the channel compression module.

3. **Importance of auxiliary data**

   Additionally, we are conducting a test to see if the auxiliary data (climate and soil) is contributing to the rice yield prediction. To check this, both the 2D CNN and 3D CNN models are implemented with and without the auxiliary data (with only S2 data).

5.2.4 Performance Evaluation

All the experiments for yield estimation are quantitatively evaluated using RMSE metrics: After the results from experiments, the scenario in which the models performed the best is selected for the final training and regression. The results are then quantitatively evaluated based on RMSE and qualitatively through the visual inspection of regression maps.

5.2.4.1 Quantitative Approach

RMSE is the most commonly used metric for this purpose and all the related works discussed in section 2.2.2 have used RMSE to compare the results between the expected and predicted yield values. The following subsection explains it briefly:

Root Mean Square Error (RMSE)

Root mean square error is a standard way to quantify the error of a model in predicting quantitative data. It is the square root of the average of squared differences between the expected and the observed values. In other words, RSME is the standard deviation of the residuals (prediction errors). It is a loss function commonly used in the regression task to verify the experimental results. The formula to calculate RMSE is shown in the equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \widehat{y_i})^2} \tag{5.6}$$

In this equation, y_i is the expected value, $\widehat{y_i}$ is the observed value and n is the number of samples. In the case of RMSE metrics, the errors are squared before they are averaged ensuring the error is always positive. For this study, RMSE is calculated in Kg/ha (kilogram per hectare) as the ground truth production data is available in the same unit.

5.2.4.2 Qualitative Approach

After obtaining the results from experiments under different scenarios, the best performing models among all scenarios are selected to generate the regression maps. The approach for generating the regression maps is similar to that of classification maps as explained in section 5.1.4.

6 RESULTS AND DISCUSSION

This chapter explains the findings of the experimental designs and performance comparison as described in Chapter 6. This chapter is also divided into two sections. The first section reports the findings of CNN design experiments with different hyperparameter combination. This is followed by a detailed explanation of the performance evaluation of different classifiers and their comparison with the existing work. This section ends with the comparison of classification maps generated by all the implemented classification models.

6.1 Classification

6.1.1 CNN Hyperparameter Sensitivity Analysis

6.1.1.1 Effect of Patch Sizes

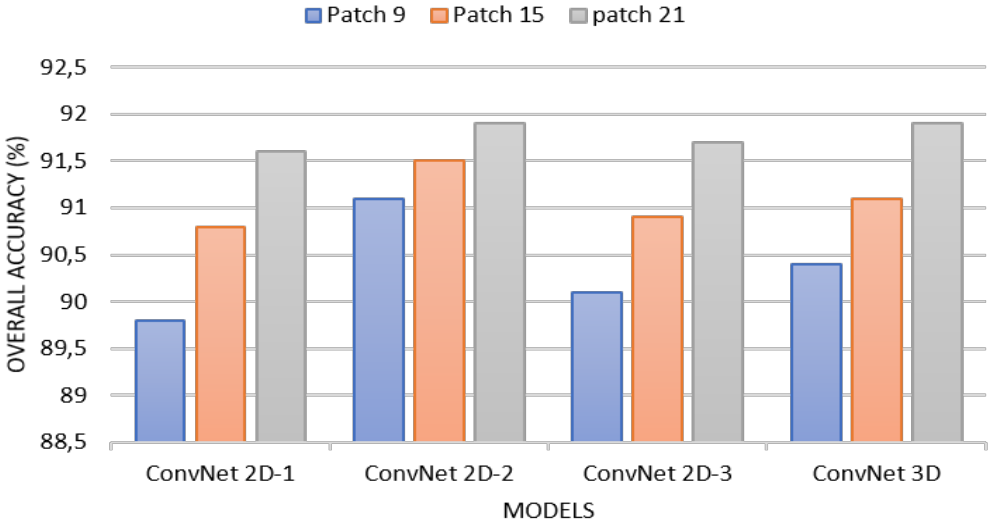

Figure 6.1: Effect of varying input patch size

Figure 6.1 illustrates how the classification performance of all the CNN models varies with the change in input patch size. While varying the input patch size in this experiment, the learning rate value was set to 0.01 and mini-batch size to 100. It is clear from the figure that the classification accuracy increased with increasing patch size and all the classifiers achieved the highest overall accuracy with patch size 21 indicating the importance of spatial context for classification. Therefore, the input patch size of 21 will be considered hereafter in all the remaining experiments.

6.1.1.2 Effect of Learning Rate

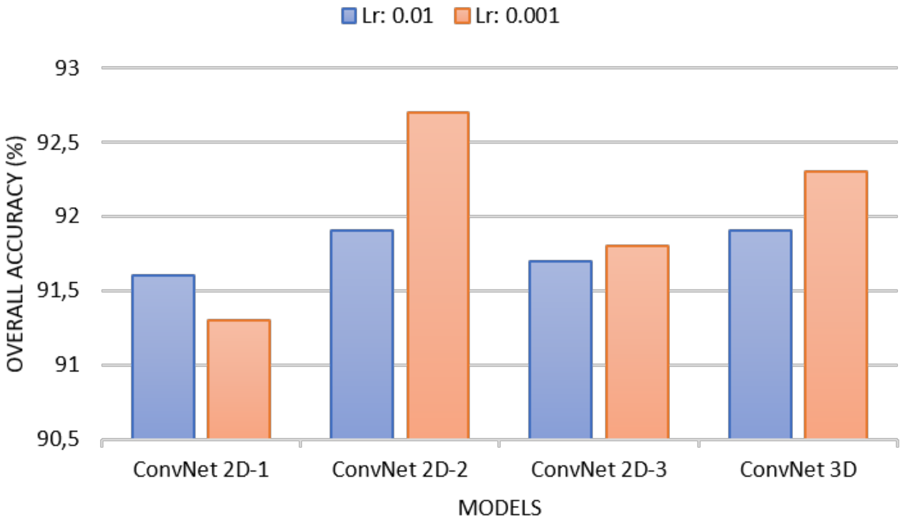

Figure 6.2: Effect of the varying learning rate

The effect on models' performance with a change in learning rate is demonstrated in Figure 6.2. In this experiment, the patch size is set to 21 and mini-batch size to 100. As a result, the first model, ConvNet 2D-1 achieved an overall accuracy of 91.6% with a learning rate of 0.01 and 91.3% with a learning rate of 0.001indicating the improved performance of the model with higher learning rate value. In contrast to this result, the remaining three models: ConvNet 2D-2, ConvNet 2D-3 and ConvNet 3D exhibited better overall accuracy of 92.7% , 91.8% , and 92.3 % respectively with the smaller learning rate of 0.001which is 0.8% , 0.1% , and 0.4% higher than that with 0.01. This result also supports the result of [33] in the case of the last two models.

The learning rate plays an important role in the network convergence in order to train a model efficiently and has a significant effect on generalization accuracy. In particular, lowering the learning rate below that which results in the fastest convergence can significantly improve the accuracy of generalization, especially when dealing with large, complex problems [55]. The experimental results on varying learning rates apparently indicate that the complex networks have better performance with a smaller learning rate. Based on the experimental results, the learning rate of 0.01 is used for ConvNet 2D-1 and 0.001 is used for the other three models in upcoming experiments.

6.1.1.3 Effect of Mini-batch Size

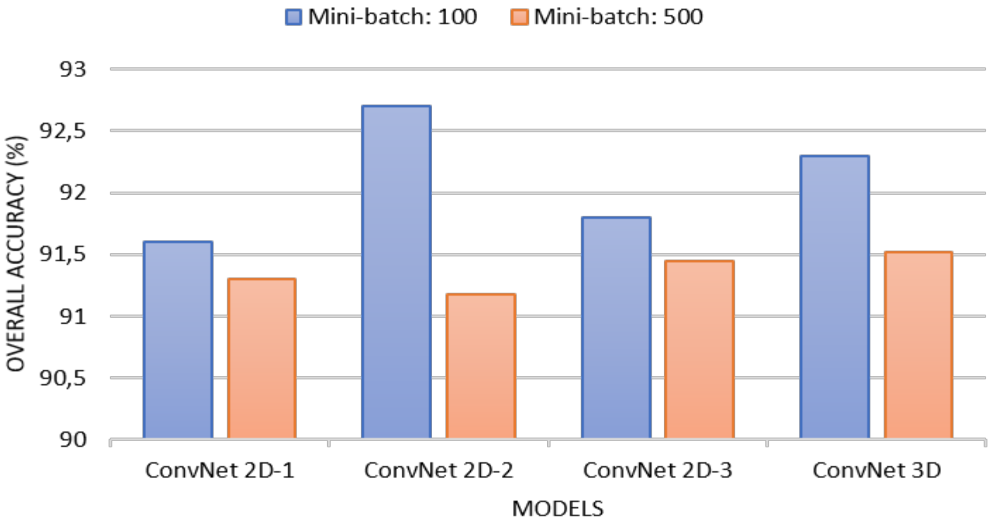

Figure 6.3: Effect of varying mini-batch size

The bar chart in Figure 6.3 represents the change in models' performance with the varying mini-batch size. It is clear from the figure that the models perform better with the mini-batch size 100 whereas the overall accuracy of all the models lowered with the mini-batch size 500. The result can be interpreted as larger mini-batch sizes tend to degrade the generalization of the classifiers which results in lower accuracy [68].

6.1.2 Significance of Multi-temporal Inputs

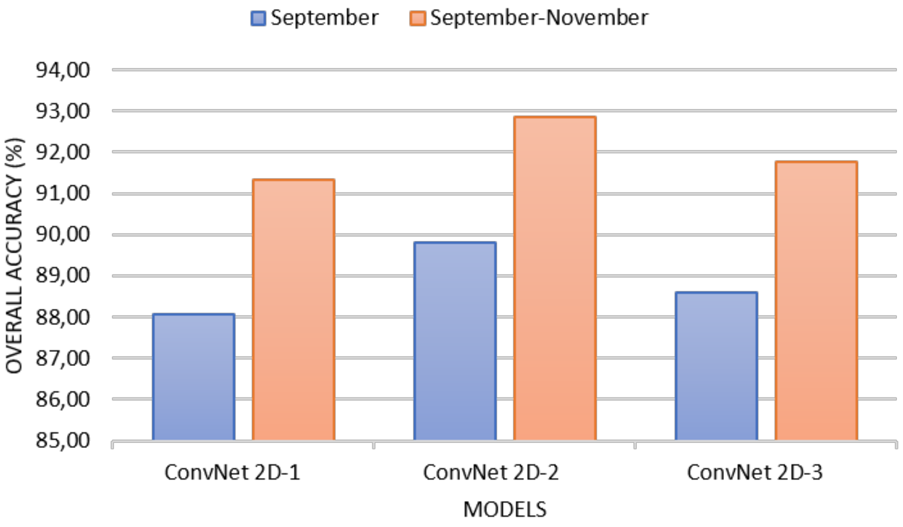

Figure 6.4: Significance of multi-temporal inputs

Figure 6.4 clearly demonstrates that the overall accuracy of all CNN-based classifiers increased significantly with the use of multi-temporal images (September-November) instead of using images of a single time period (September). Since the fourth model, ConvNet 3D requires input images of more than one time period, only 2D CNN-based models are compared in this experiment.

6.1.3 Performance Evaluation of Classifiers

After the hyperparameters tuning experiments, each model with the best hyperparameter combination was implemented to classify the rice crop for each year. Along with the four patch-based CNN architectures as explained in section 5.1.1, pixel-based SVM is employed as a baseline for rice crop classification. To evaluate the performance of all the classifiers two metrics namely overall accuracy and F1-score were used. As explained in section 5.1.2 in detail, a one-tile-out scheme for three random tiles (T45RVK, T45RUK, and T44RQR) is used to generate an average classification result for the all the classification models in all the years considered (2016-2018).

Table 6.1 shows the classification result of all the classifiers including SVM for the years 2016-2018. For instance, if we see the classification results of 2016 in the table, it clearly depicts that all the CNN-based models have outperformed the performance of the classical SVM classifier. The pixel-based SVM classifier achieved an overall accuracy of 81.47% which on average is over 10% lower than all the CNN models. Similarly, in the same ratio, the values of precision, recall, and F1-score for rice class (the class we are interested in) are lesser than that of all CNN-based models. Among the chosen CNN classifiers, ConvNet 2D -2 showed the best performance with the highest overall accuracy if 91.77%. The second-best model is ConvNet 3D which is followed by ConvNet 2D-3 with the overall accuracy of 91.77% and 92% respectively. Among the CNN classifiers, the lowest classification accuracy (91.33%) was obtained from the first model i.e. ConvNet 2D-1. Furthermore, if we see the classification results of the remaining years, we can observe the same pattern in the accuracy achieved by all the tested models in terms of both specific and relative accuracies. The consistency of the results reveals that the CNN approach is able to better generalize across the considered years.

The classification results averaged for three considered years are shown in Table 6.2. As the classification results are consistent in all these years, the averaged result is in line with the result of 2016. This means that the least accuracy is obtained by SVM with an average overall accuracy of 81.66% . This result supports the findings from the previous studies that CNN-based models are able to extract the features and classify the rice pixels way better than the classical SVM approach. Thereafter, ConvNet 2D-1 has the lowest overall accuracy of 91.23% among all the CNN models. The reason behind the consistent lowest performance of this model can be justified by the simplicity of the network architecture which is not able to accurately learn the features and optimally classify the rice pixels.

		Year 2016			
Model	**Class**	**Precision**	**Recall**	**F1**	**OA**
SVM	Rice	0.75	0.72	0.73	81.47
	Non-rice	0.85	0.90	0.86	
ConvNet 2D-1	Rice	0.87	0.88	0.88	91.33
	Non-rice	0.94	0.93	0.93	
ConvNet 2D-2	Rice	0.90	0.88	0.89	92.87
	Non-rice	0.93	0.95	0.94	
ConvNet 2D-3	Rice	0.87	0.90	0.89	91.77
	Non-rice	0.94	0.93	0.94	
ConvNet 3D	Rice	0.88	0.90	0.89	92.00
	Non-rice	0.94	0.93	0.94	

		Year 2017			
Model	**Class**	**Precision**	**Recall**	**F1**	**OA**
SVM	Rice	0.76	0.77	0.77	81.63
	Non-rice	0.85	0.85	0.85	
ConvNet 2D-1	Rice	0.86	0.92	0.89	91.40
	Non-rice	0.94	0.91	0.93	
ConvNet 2D-2	Rice	0.90	0.94	0.91	93.57
	Non-rice	0.96	0.93	0.95	
ConvNet 2D-3	Rice	0.88	0.90	0.89	91.73
	Non-rice	0.94	0.93	0.93	
ConvNet 3D	Rice	0.88	0.92	0.90	91.93
	Non-rice	0.95	0.92	0.93	

		Year 2018			
Model	**Class**	**Precision**	**Recall**	**F1**	**OA**
SVM	Rice	0.78	0.77	0.78	81.87
	Non-rice	0.85	0.85	0.85	
ConvNet 2D-1	Rice	0.88	0.92	0.90	90.97
	Non-rice	0.93	0.91	0.92	
ConvNet 2D-2	Rice	0.90	0.94	0.92	92.90
	Non-rice	0.95	0.92	0.94	
ConvNet 2D-3	Rice	0.88	0.92	0.90	91.30
	Non-rice	0.94	0.91	0.92	
ConvNet 3D	Rice	0.89	0.92	0.90	91.83
	Non-rice	0.94	0.92	0.93	

Table 6.1: Classification results in terms of precision, recall, F1-score, and overall accuracy for years 2016-2018

Model	Class	Precision	Recall	F1	OA
SVM	Rice	0.76	0.75	0.76	81.66
	Non-rice	0.85	0.87	0.85	
ConvNet 2D-1	Rice	0.87	0.91	0.89	91.23
	Non-rice	0.94	0.92	0.93	
ConvNet 2D-2	Rice	0.90	0.92	0.91	93.11
	Non-rice	0.95	0.94	0.94	
ConvNet 2D-3	Rice	0.88	0.91	0.89	91.60
	Non-rice	0.94	0.92	0.93	
ConvNet 3D	Rice	0.88	0.91	0.90	91.92
	Non-rice	0.94	0.93	0.93	

Table 6.2: Average classification results in terms of precision, recall, F1-score, and overall accuracy

Subsequently, if we see the classification accuracies of ConvNet 2D-3 and ConvNet3D, the performance of the later is better than the former. The average overall accuracy obtained by ConvNet 3D is 91.92% which in the case of ConvNet 2D-3 is 91.60% . As explained in 5.1.1 that the network architecture of these two models is exactly the same except the input data representation, it could be inferred that the improved performance of the ConvNet 2D-3 is contributed by the temporal features extracted by the 3D CNN-based model. This result also supports the findings of [33] from which both the network architecture is adopted.

Among all these classifiers, ConvNet 2D-2 achieved the highest accuracy in terms of all the metrics (precision, recall, F1-score, and overall accuracy). The average overall accuracy obtained by this model is 93.11% which is 1.19% higher than ConvNet3D. While the 3D CNN-based model is expected to perform the best in theory and from previous studies, in this case, the 2D CNN-based model, ConvNet 2D-2 contrasts the expectation by demonstrating the better result. This can be justified by two reasons. Firstly, in contrast to the previous studies which dealt high temporal dimension of the data, the limited temporal data of two time period is not able to fully exploit the potential of 3D CNN model. Secondly, the network architecture of ConvNet 2D-2 (see Figure 5.2) consists of a large number of filters as compared to ConvNet3D and apparently with a higher number of filters, the network is able to extract a higher number of extractions from image data helping in the better performance of the model.

6.1.4 Comparison with the existing works

Besides the performance evaluation of the implemented networks and comparison of the resulting accuracy among them, further, exploration is done in an illustrative way, with the results from previous studies which have implemented the adopted network architecture with different dataset in a different study area. This is done to check the suitability of Sentinel-2 for crop classification, in particular, rice crop in this case, in the study area. The overall accuracy of the implemented models and that from the existing works are listed in Table 6.3 for comparison. [27] showed that it is possible to obtain an average OA of

	Zhang et al. [27]	Ji et al. [33]		
Model	**Landsat-8 (30m)**	**GF1 (15m)**	**GF2 (4m)**	**Sentinel-2 (20m)**
ConvNet 2D-2	91.3%	-	-	93.11 %
ConvNet 2D-3	-	77.2%	95.6%	91.60%
ConvNet 3D	-	79.4%	96.8%	91.92%

Table 6.3: Comparison of classification accuracy with existing works

91.23% when using Landsat-8 to identify rice crops. The network architecture adopted from this study is ConvNet 2D-2 which demonstrated the highest classification accuracy with an average OA of 93.11% . The increased accuracy can be attributed to the higher spatial resolution of S2 in comparison to the medium resolution of Landsat which adds motivation to conduct further relevant studies in the future using S2 data.

In [33], the authors achieved OA of 77.2% and 79.4% with 2D CNN and 3DCNN models respectively using Gaofen1(GF1) images with a spatial resolution of 15m. Furthermore, with the use of Gaofen2 (GF2) images with a high spatial resolution of 4m, the study achieved a very high average OA of 95.6% and 96.8% with 2D CNN and 3DCNN models respectively. In comparison to these results, the models ConvNet 2D-3 and ConvNet3D adopted from this study achieved an average OA of 91.60% and 91.92% respectively. On the one hand, our results demonstrate much higher accuracy of S2 images with 20m spatial resolution in comparison to the results obtained by [33] with GF1 images of 15m spatial resolution and relatively lower accuracy in comparison to the results with GF2 images of 4m spatial resolution. On the other hand, the 3D CNN employed by [33] contributes an additional accuracy of 2.2% and 1.2% with GF1 and GF2 images respectively which in our case is an average of 0.32% with S2 dataset. As explained earlier, the reason behind this relatively lower contribution of the 3D CNN architecture is the limited temporal data available for the study besides the resolution of the S2 data and new study area where the models are implemented. In summary, the comparison of the classification results obtained from this study with the existing works clearly suggests the suitability of S2 MSI data, with a high spatial and temporal resolutions, for rice crop mapping in developing countries like Nepal.

6.1.5 Comparison of Classification Maps

An example of rice maps generated by all the models that we implemented, and the corresponding ground-truth maps are shown in Figure 6.5. The generated maps are of tile T45RUK in the year 2016. The green pixels in the classified maps represent rice, brown pixels signify non-rice and white pixels are those which were filtered with the NDVI threshold and include non-vegetation areas and clouds.

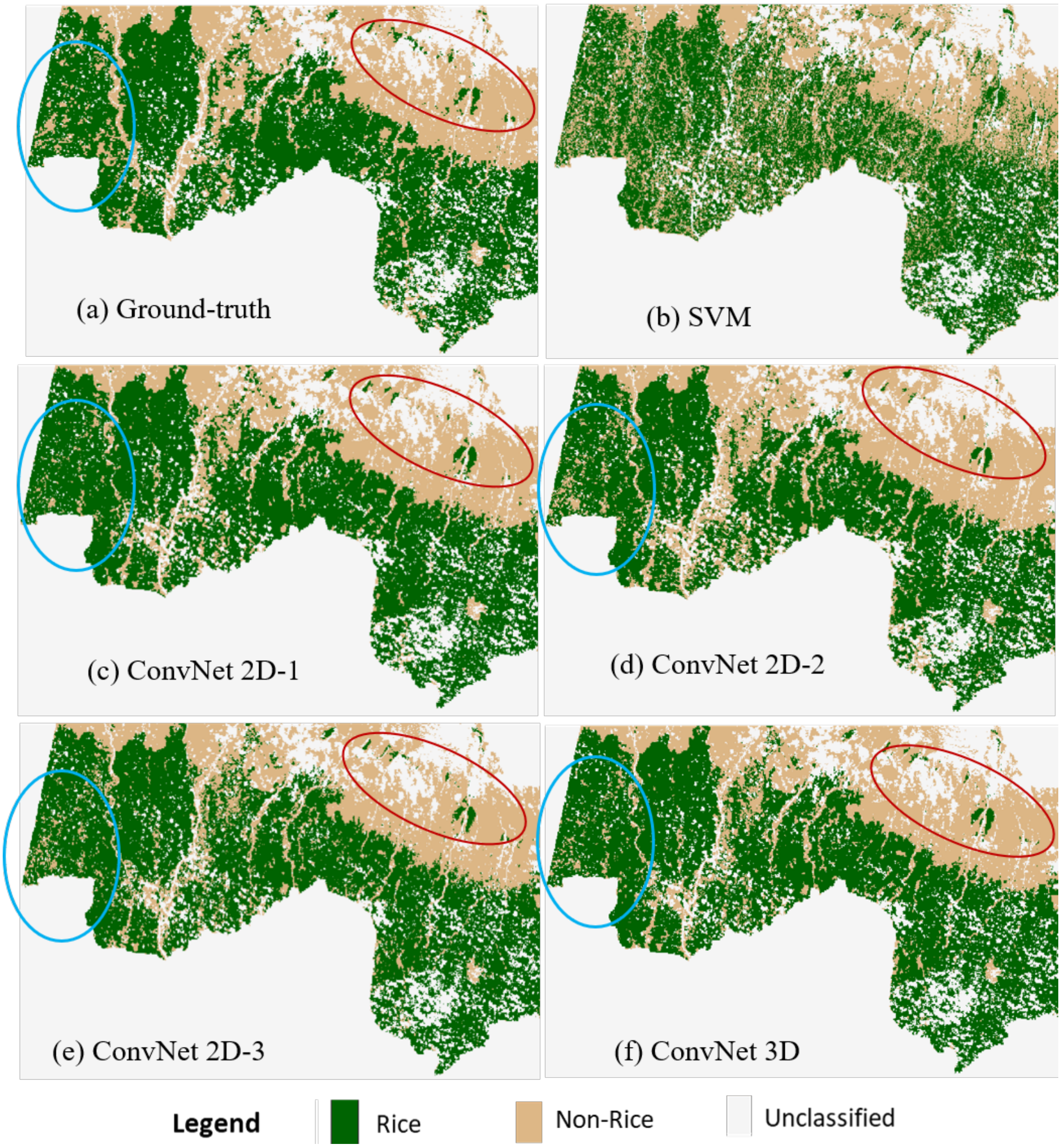

Figure 6.5: (a) Ground truth-map of T45RUK tile in 2016; Rice maps generated by (b) SVM, (c) ConvNet 2D-1, (d) ConvNet 2D-2, (e) ConvNet 2D-3, and (f) ConvNet 3D (Blue and red circle represent exemplary areas used for discussing visual difference in model performance)

From a qualitative perspective, we can visually compare the rice maps generated by the CNN models and SVM against the ground truth map. For instance, if we compare Figure 6.5 (a) with Figure6.5 (b), we can see that the rice map generated by SVM is not smooth and has salt-and-pepper noise. Unlike the result from SVM, the rice maps generated by the CNN-based models are smoother and resemble the ground-truth map to a great extent in a broad view. At a closer look, we can notice some areas where the predicted labels differ from the ground-truth map. For example, ConvNet 2D-1 (Figure 6.5(c)) has overpredicted the rice pixels in an area highlighted with a blue circle in comparison to the maps from other models while the map from ConvNet 2D-2 in that area shows the most match with the ground-truth map. Similarly, in the area highlighted with a red circle, all the models have underpredicted the rice pixels especially, ConvNet-2D. This could be because of the presence of noise in the image (white pixels) which is impacting the prediction of the CNN models. Overall, the qualitative analysis through the visual comparison of the maps depicts that the rice maps generated by CNN models resemble the ground-truth, but it is difficult to compare the performance of the models visually.

6.2 Yield Estimation

6.2.1 Experiment 1

As we have mentioned earlier that we have limited time-series S2 data and sparse ground-truth in the study area, in the first experiment, we are validating the performance of the CNN networks in this scenario. For this, the whole dataset from 3 years is randomly divided into training (70%) and validation (30%) dataset without considering the extent of S2 tiles or district boundaries.

1. **Variation in ground-truth labels**

   To use ground-truth yield data for training the network, in the first case, pixel level yield (kg/pixel) is computed from district level yield (kg/ha). This results in the uniform ground-truth labels (GT labels) for each district without any variation. While in the second case, the resulting GT labels are varied by 10% in proportional to the corresponding NDVI value of S2 images in the peak of the season. The network performance of both the models was validated with these two types of GT labels and the result is shown in Figure 6.6.

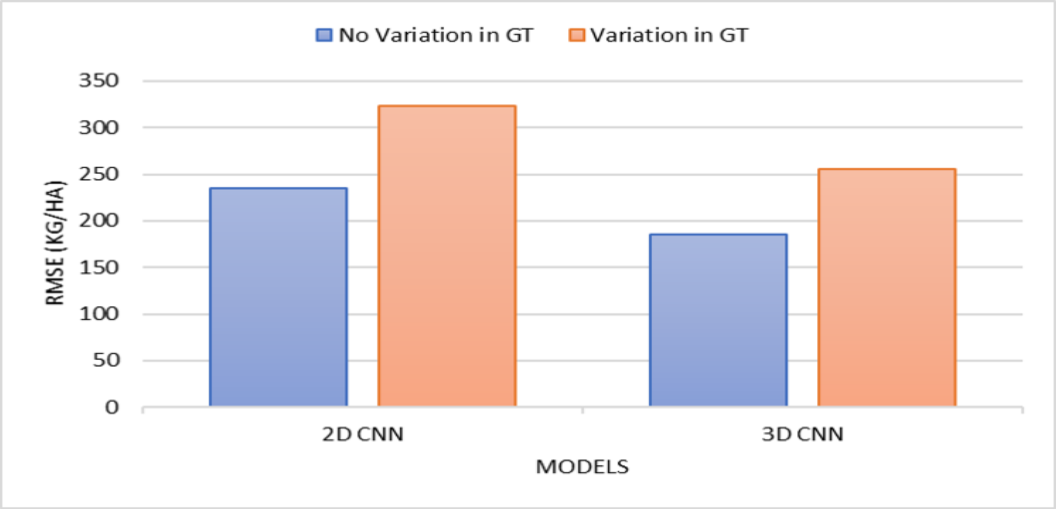

Figure 6.6: Effect of variation in the ground-truth labels (model validation)

   Figure 6.6 reveals that both the models performed better without GT labels variation with lower RMSE than that with variations in the labels. This indicates that introducing variation in the input labels is not helping in the improvement of the network performance.

2. **Testing the significance of channel compression module in 3DCNN**

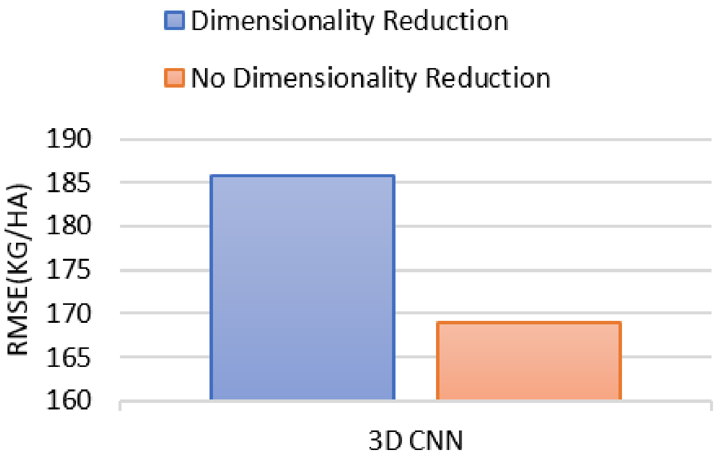

Figure 6.7: Effect of dimensionality reduction (model evaluation)

As mentioned earlier in section 6.1.2, an additional test was done to check if the dimensionality reduction technique from 20 channels to 3 is contributing to the performance of the 3D CNN model. The result of this test as shown in 6.7 clarifies that the model performed better without the dimensionality reduction technique with reduced RMSE from 186kg/ha to 169 kg/ha. This suggests the use of all the channels during network training instead of projecting the number of channels from 20 to 3 so as to improve the performance of the model. Figure 6.8 shows the 3D CNN architecture after introducing a variation by removing the dimensionality reduction module.

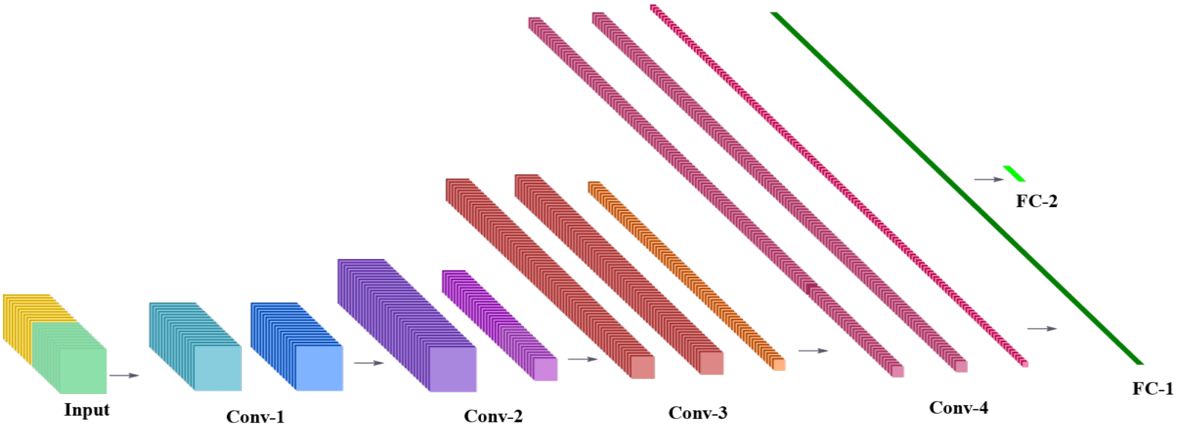

Figure 6.8: Variation in 3D CNN architecture by removing dimensionality reduction module

3. **Importance of auxiliary data**

The last test was conducted to verify if the auxiliary data (climate and soil data) is providing an additional contribution to the model performance in the yield prediction process. To verify this, the models' performances were evaluated with and without auxiliary data and the result is illustrated in Figure 6.9. It seems evident from the result that the use of auxiliary data in the yield estimation process has an added contribution resulting in improved performance.

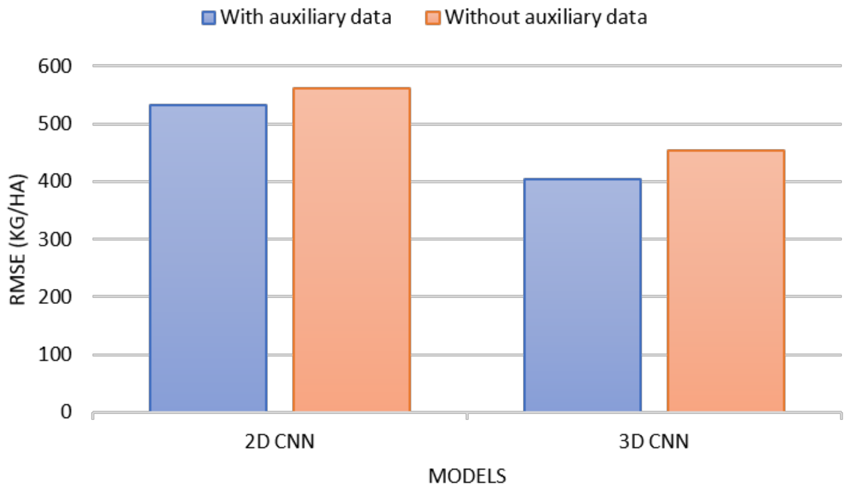

Figure 6.9: Contribution of auxiliary data in yield estimation (model validation)

6.2.2 Experiment 2

The results from experiment 1 make it clear that both the models are able to estimate the rice yield by with low RMSE. This indicates that the model is performing well with the S2 data along with auxiliary data in the study area. After the model validation, in experiment 2, we are evaluating the performance of the model with 3 test tiles that cover 5 districts. These test tiles are from years 2016-2018 so, the models will predict the average yield and calculate the mean RMSE. Similar to the tests in experiment 1, we have conducted the tests to evaluate the model performance in three different scenarios as explained below:

1. **Variation in ground-truth labels**

   In line with the result in experiment 1, both the model performed better without variation in GT-labels. However, it is important to notice that RMSE values are relatively higher in both models indicating that the models are not able to generalize properly in the new dataset. Moreover, a simple 2D CNN is performing better than a complex 3D CNN model which is exactly opposite to our expectation. The reason behind this is clarified in the following test.

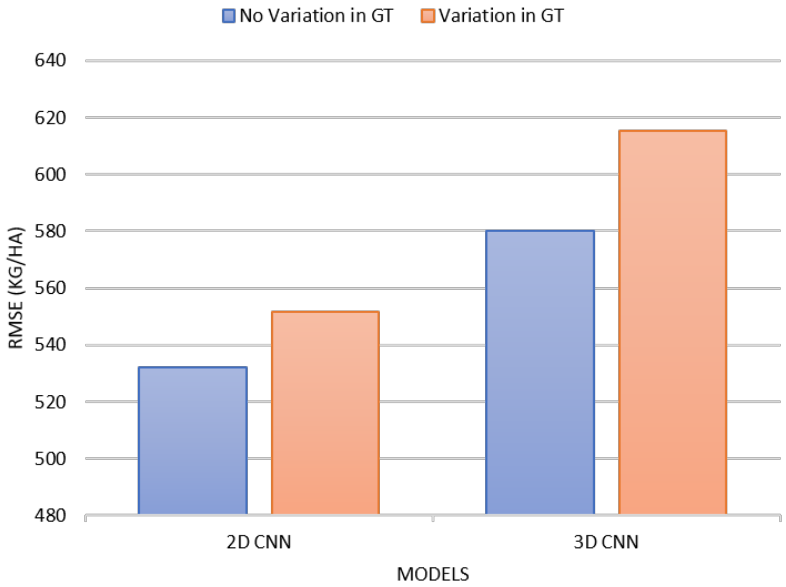

Figure 6.10: Effect of variation in the ground-truth labels (model evaluation)

2. **Testing the significance of channel compression module in 3DCNN**

   By removing the dimensionality reduction module form the network architecture, RMSE of 3D CNN reduced significantly from 580 kg/ha to around 400kg/ha. With this, the performance of 3D CNN got improved than 2D CNN (530 kg/ha) as well. Therefore, it is clear that in the initial experiment, the 3D CNN was performing worst because the channels were projected before the network training process which lowered the model performance.

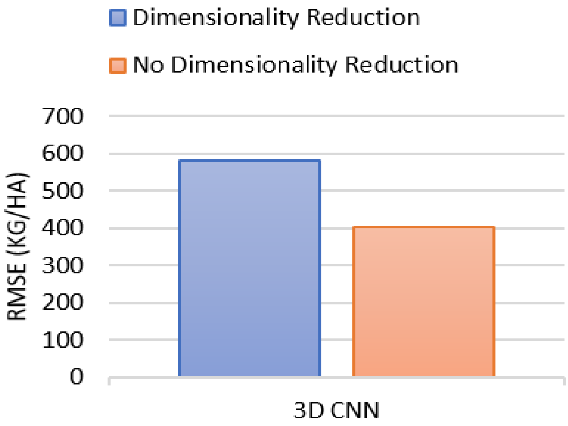

Figure 6.11: Effect of dimensionality reduction (model evaluation)

3. **Importance of auxiliary data**

   The last test to verify the significance of auxiliary data in the yield estimation process in the test dataset (Figure 6.12), indicates clearly that both the CNN models performed better with lower RMSE with the use of auxiliary data besides the S2 data.

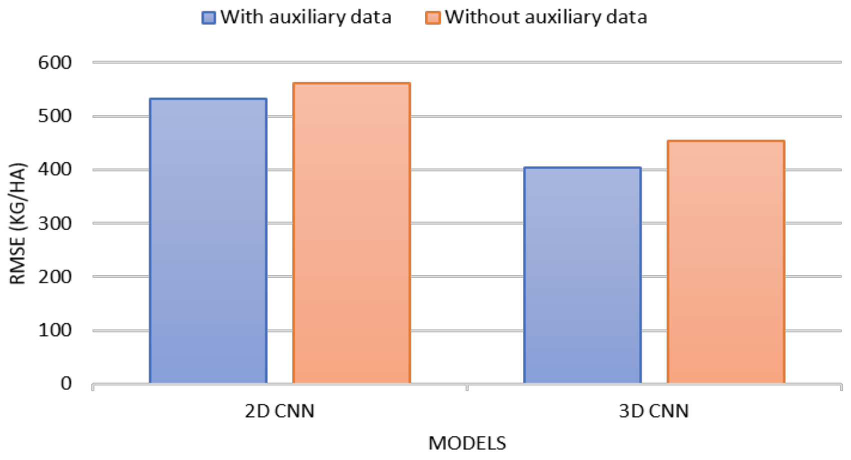

Figure 6.12: Contribution of auxiliary data in yield estimation (model evaluation)

6.2.3 Performance Evaluation of CNN-based Regression Models

Based on the results obtained from the experiments conducted with different scenarios, the best result obtained for each model was considered. To summarize, the 2D CNN model performed best without GT labels variation and with auxiliary data. In addition to these scenarios, 3D CNN demonstrated the highest performance without dimensionality reduction module in the network architecture. Besides these CNN models for regression, a linear SVR was employed as a baseline algorithm for yield estimation with pixel-based data input. Table 6.3 shows the RMSE values obtained from all the models while estimating the rice yield.

The results clearly depict that both CNN models performed better than SVR and 3D CNN achieved the lowest RMSE signifying the best performance in the yield estimation process. Obviously, all the models obtained significantly lower RMSE in experiment 1 in which the validation dataset is randomly taken from the training dataset. While in experiment 2, the RMSE of models is higher since we are evaluating the performance of the model in a completely new area. Despite the differences, the performance of 3D CNN in experiment 2 is remarkable considering the limitations in the data. Considering average yield per hectare of districts, this RMSE value of 3D CNN represents around 5% error in experiment 1 and around 11% error in experiment 2 in the estimated yield.

6.2.4 Comparison with the existing works

	(kg/ha)	
Models	**Experiment 1**	**Experiment 2**
SVR	357.23	609.12
2D CNN	235.5	551.75
3D CNN	169	404.25

Table 6.4: Results of SVR, 2D CNN, and 3D CNN in terms of RMSE (kg/ha)

The author in [13] predicted soybean yield in the U.S by using long term MODIS dataset (13 years) for training the 3D CNN model and evaluated its performance with test dataset of a year. This study achieved recommendable results with an average RMSE of 5.27 bushels per acre which is around 355 kg/ha. While we adopted the same network architecture of 3D CNN for this study in estimating rice yield, the network is implemented in a different study area, using the Sentinel-2 dataset of 3 years with a temporal depth of 2 times per year. Moreover, we have introduced a variation in the 3D CNN network by avoiding the dimensionality reduction technique. While it is obvious to get better accuracy in experiment 1, the result validates the model performance in our scenario. In the second experiment which is more practical in terms of implementation, RMSE of 404.25 kg/ha is obtained which is quite high in comparison to the result from [13] with RMSE of 355 kg/ha. However, considering the limitations in the temporal dataset and the use of sparse ground-truth data, the results obtained from the model are still remarkable.

6.2.5 Comparison of Regression Maps

Figure 6.13 shows an example of maps illustrating the yield estimated by (b) SVR, (c) 2D CNN, and (d) 3D CNN and the corresponding ground-truth map (a). Variation in yield is represented by shades of green color where dark green represents high yield values and the lighter shade represents low yield. The district boundary is shown in black color. Similar to classification maps, white pixels indicate no data areas and brown color represents no production areas. If we see the ground-truth map (Figure 6.13 (a)), variation in yield per district is visible. In comparison to the ground-truth, the output from SVR shows that yield is overestimated resulting in the highest RMSE value as shown in Table 6.4. On the contrary, the maps generated by CNN models show more similarity to the ground-truth map. However, at a closer look, we can see some differences. For instance, in the area highlight by a red circle in Figure 6.13, the regression map from 2D CNN shows slightly overestimation in yield that that from 3D CNN as compared to the ground-truth. In a broader look, with visual maps, it is difficult to evaluate the performance of models precisely.

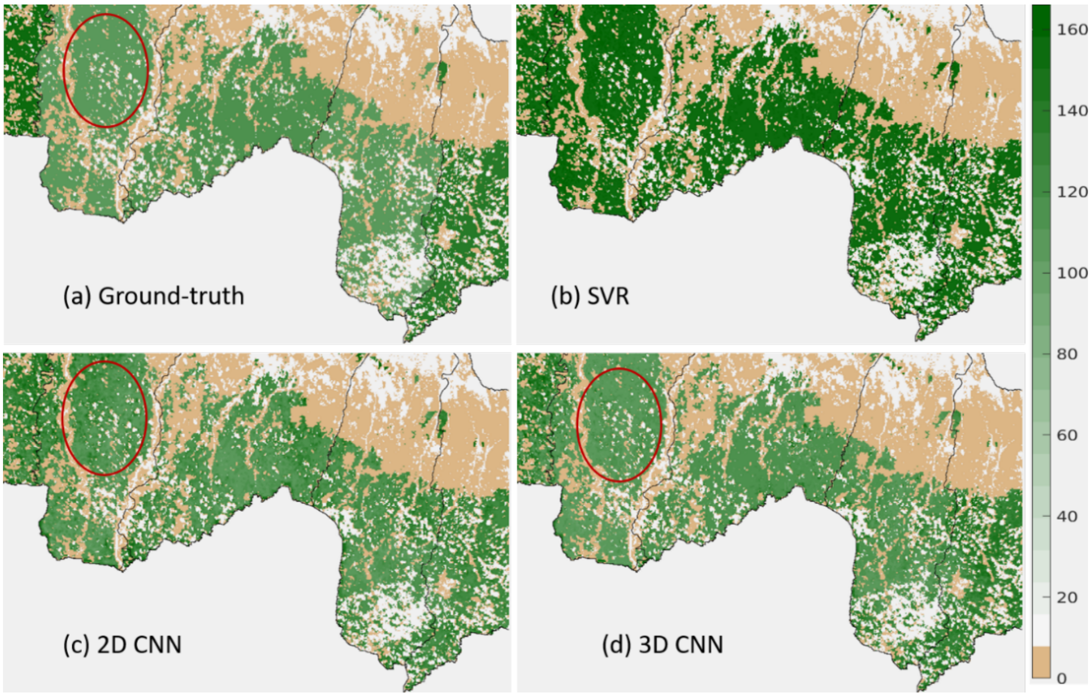

Figure 6.13: (a) Ground-truth map of T45RUK tile; Maps showing estimated yields, generated by (b) SVR, (c) 2D CNN, and (d) 3D CNN

7 CONCLUSION AND FUTURE WORKS

7.1 Conclusion

This thesis presents the use of multitemporal Sentinel-2 images for rice crop classification and yield estimation using the deep learning framework with a focus on developing countries. The feasibility of this work was tested through a case study in Terai Districts of Nepal which is one of the developing countries. The results demonstrate the viability of S2 images for rice crop classification and yield estimation in developing countries where data is scarce, and ground-surveys are expensive and time-consuming.

The conclusions of this thesis are presented based on the objectives which are:

1. *To review existing state-of-art deep learning algorithms for rice crop classification and yield estimation and select the suitable architectures.*

   In Chapter 2, existing deep learning algorithms for crop classification and yield estimation using multi-temporal satellite imagery were reviewed. After exploring the trend of deep learning approaches used in this field, CNN was chosen as a core algorithm because of its remarkable performance in previous works. For classification, we employed four patch-based CNN models namely **ConvNet 2D-1, ConvNet 2D-2** [27]**, ConvNet 2D-3, and ConvNet 3D** [33] to investigate the difference in model performance with the change in network architecture. Similarly, for yield estimation, we employed **2D CNN and 3D CNN [13]** models. Furthermore, **SVM and SVR** were employed as the baseline algorithm.

2. *To implement and optimize the performance of the chosen architectures*

   To fulfill the second objective, all the chosen CNN models and baseline algorithms were implemented in Chapter 5. The models' sensitivity to hyperparameters were tested to select the optimal hyperparameter values and enhance model performance. In addition to this, several experiments were conducted considering different scenarios of data input to the CNN models.

   - By performing experiments to investigate the significance of multi-temporal data in rice classification, we conclude that the use of multi-temporal data improves the classification accuracy as opposed to the use of images of a single time period.

- By performing the experiments to explore the implication of the use of auxiliary data (climate and soil) on the yield estimation process, we conclude that auxiliary data add a contribution to the model performance resulting in better yield estimation.
- By performing the experiment to investigate the role of channel compression module in yield estimation network architecture, we conclude that channel compression at the beginning reduces the model performance and it is advised to consider all the available spectral bands to enhance the learning process of models and improve the results.

3. *To evaluate the performance of classification and yield estimation models using performance metrics and validate their performance with reference to the existing works..*

   The evaluation of classification models based on overall accuracy and F1-score statistics demonstrate that the CNN models outperformed the SVM approach of classification. Among the CNN models, ConvNet 2D-2 obtained the highest classification accuracy followed in order by ConvNet 3D, ConvNet 2D-3 and ConvNet 2D-1. From the results, we can infer two things: firstly, in case of similar network architecture, 3D CNN performed better than 2D CNN confirming that spatio-temporal representations in 3D CNN improve the classification accuracy. Secondly, the better performance of ConvNet 2D-2 indicates that the complexity of network architecture with a large number of filters is contributing to increased accuracy of the model and also suggests the 3D form of this model could result in the best accuracy among all. Similarly, the evaluation of yield estimation models based on root means square error showed that 3D CNN models outperformed the basic 2D CNN and SVR.

   In the last phase, the reliability of the performance of the implemented models was checked with reference to the accuracy obtained in the corresponding works from which the network architectures were adopted. The comparison shows that we have obtained remarkable accuracy in both rice classification and yield estimation process suggesting the feasibility of both dataset and approach for crop monitoring process in developing countries as well.

With this, we have successfully achieved our aim which is "*to support crop monitoring process by investigating the viability of Sentinel-2 data for rice crop classification and crop yield estimation in the Terai region of Nepal using deep learning approach*" . We forsee that deep learning-based crop classification and yield estimation using freely available Sentinel-2 imagery will contribute to more efficient crop growth monitoring and managing agriculture practices in the future.

7.2 Future Works

The results presented in chapter 6 have room for improvement with more possible scenarios to be explored. Following are the list of recommended works for future as identified by the author:

- Image fusion techniques can be considered to fill the data gaps due to significant cloud cover and to reduce the effect of data noise in model performance.
- Increasing per year temporal depth of S2 images can be advantageous in both the classification and yield estimation process. However, we have to keep in mind that this will also increase the data volume and computational costs especially when the study area is large.
- Considering the limited data availability for the yield estimation process, the use of pre-trained networks, if available in this scenario, can be explored to see if they improve the results.
- Successful results from the use of the multi-receptive-field network in the super-resolution field suggest that it could be advantageous in this field as well. So, this approach can be explored in further studies.
- Exploration of deeper CNN architectures like residual network (ResNet) and Inception net in this field is recommended since it has not been done yet.

Bibliography

[1] R. GEBBERS and V. I. ADAMCHUK. “Precision agriculture and food security”. In: *Science* 327.5967 (2010), pp. 828–831.

[2] U. NATIONS. “Transforming our world: The 2030 agenda for sustainable development”. In: *New York: United Nations, Department of Economic and Social Affairs* (2015).

[3] N. MINAMIGUCHI. “The application of geospatial and disaster information for food insecurity and agricultural drought monitoring and assessment by the FAO GIEWS and Asia FIVIMS”. In: *Workshop on Reducing Food Insecurity Associated with Natural Disasters in Asia and the Pacific*. Vol. 27. 2005, p. 28.

[4] J. DONG, X. XIAO, W. KOU, Y. QIN, G. ZHANG, L. LI, C. JIN, Y. ZHOU, J. WANG, C. BIRADAR, et al. “Tracking the dynamics of paddy rice planting area in 1986–2010 through time series Landsat images and phenology-based algorithms”. In: *Remote Sensing of Environment* 160 (2015), pp. 99–113.

[5] Y. ZHOU, X. XIAO, Y. QIN, J. DONG, G. ZHANG, W. KOU, C. JIN, J. WANG, and X. LI. “Mapping paddy rice planting area in rice-wetland coexistent areas through analysis of Landsat 8 OLI and MODIS images”. In: *International journal of applied earth observation and geoinformation* 46 (2016), pp. 1–12.

[6] D. PENG, A. R. HUETE, J. HUANG, F. WANG, and H. SUN. “Detection and estimation of mixed paddy rice cropping patterns with MODIS data”. In: *International Journal of Applied Earth Observation and Geoinformation* 13.1 (2011), pp. 13–23.

[7] M. IMMITZER, F. VUOLO, and C. ATZBERGER. “First experience with Sentinel-2 data for crop and tree species classifications in central Europe”. In: *Remote Sensing* 8.3 (2016), p. 166.

[8] J. INGLADA, M. ARIAS, B. TARDY, O. HAGOLLE, S. VALERO, D. MORIN, G. DEDIEU, G. SEPULCRE, S. BONTEMPS, P. DEFOURNY, et al. “Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery”. In: *Remote Sensing* 7.9 (2015), pp. 12356–12379.

[9] Y. CAI, K. GUAN, J. PENG, S. WANG, C. SEIFERT, B. WARDLOW, and Z. LI. “A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach”. In: *Remote sensing of environment* 210 (2018), pp. 35–47.

[10] F. M. QAMER, S. P. SHAH, M. MURTHY, T. BAIDAR, K. DHONJU, and B. G. HARI. "Operationalizing crop monitoring system for informed decision making related to food security in Nepal". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.8 (2014), p. 1325.

[11] J. YOU, X. LI, M. LOW, D. LOBELL, and S. ERMON. "Deep gaussian process for crop yield prediction based on remote sensing data". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[12] A. KAMILARIS and F. X. PRENAFETA-BOLDÚ. "Deep learning in agriculture: A survey". In: *Computers and electronics in agriculture* 147 (2018), pp. 70–90.

[13] H. RUSSELLO. "Convolutional neural networks for crop yield prediction using satellite images". In: *IBM Center for Advanced Studies* (2018).

[14] T. SAKAMOTO, M. YOKOZAWA, H. TORITANI, M. SHIBAYAMA, N. ISHITSUKA, and H. OHNO. "A crop phenology detection method using time-series MODIS data". In: *Remote sensing of environment* 96.3-4 (2005), pp. 366–374.

[15] H. TIAN, M. WU, L. WANG, and Z. NIU. "Mapping early, middle and late rice extent using sentinel-1A and Landsat-8 data in the poyang lake plain, China". In: *Sensors* 18.1 (2018), p. 185.

[16] Q. ZHAO, V. LENZ-WIEDEMANN, F. YUAN, R. JIANG, Y. MIAO, F. ZHANG, and G. BARETH. "Investigating within-field variability of rice from high resolution satellite imagery in Qixing Farm County, Northeast China". In: *ISPRS International Journal of Geo-Information* 4.1 (2015), pp. 236–261.

[17] Ç. KÜÇÜK, G. TAŞKIN, and E. ERTEN. "Paddy-rice phenology classification based on machine-learning methods using multitemporal co-polar X-band SAR images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.6 (2016), pp. 2509–2519.

[18] F. DODDS and J. BARTRAM. *The water, food, energy and climate Nexus: Challenges and an Agenda for action*. Routledge, 2016.

[19] M. G. CASTRO GOMEZ. "Joint use of Sentinel-1 and Sentinel-2 for land cover classification: A machine learning approach". In: *Lund University GEM thesis series* (2017).

[20] N. KUSSUL, M. LAVRENIUK, S. SKAKUN, and A. SHELESTOV. "Deep learning classification of land cover and crop types using remote sensing data". In: *IEEE Geoscience and Remote Sensing Letters* 14.5 (2017), pp. 778–782.

[21] R. SONOBE, Y. YAMAYA, H. TANI, X. WANG, N. KOBAYASHI, and K.-i. MOCHIZUKI. "Assessing the suitability of data from Sentinel-1A and 2A for crop classification". In: *GIScience & Remote Sensing* 54.6 (2017), pp. 918–938.

[22] K. UDDIN, H. L. SHRESTHA, M. MURTHY, B. BAJRACHARYA, B. SHRESTHA, H. GILANI, S. PRADHAN, and B. DANGOL. “Development of 2010 national land cover database for the Nepal”. In: *Journal of environmental management* 148 (2015), pp. 82–90.

[23] N. TORBICK, D. CHOWDHURY, W. SALAS, and J. QI. “Monitoring rice agriculture across myanmar using time series Sentinel-1 assisted by Landsat-8 and PALSAR-2”. In: *Remote Sensing* 9.2 (2017), p. 119.

[24] A. O. ONOJEGHUO, G. A. BLACKBURN, Q. WANG, P. M. ATKINSON, D. KINDRED, and Y. MIAO. “Mapping paddy rice fields by applying machine learning algorithms to multi-temporal Sentinel-1A and Landsat data”. In: *International journal of remote sensing* 39.4 (2018), pp. 1042–1067.

[25] N. GANDHI, L. J. ARMSTRONG, O. PETKAR, and A. K. TRIPATHY. “Rice crop yield prediction in India using support vector machines”. In: *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE. 2016, pp. 1–5.

[26] N.-T. SON, C.-F. CHEN, C.-R. CHEN, and V.-Q. MINH. “Assessment of Sentinel-1A data for rice crop classification using random forests and support vector machines”. In: *Geocarto international* 33.6 (2018), pp. 587–601.

[27] M. ZHANG, H. LIN, G. WANG, H. SUN, and J. FU. “Mapping Paddy Rice Using a Convolutional Neural Network (CNN) with Landsat 8 Datasets in the Dongting Lake Area, China”. In: *Remote Sensing* 10.11 (2018), p. 1840.

[28] L. ZHONG, L. HU, and H. ZHOU. “Deep learning based multi-temporal crop classification”. In: *Remote sensing of environment* 221 (2019), pp. 430–443.

[29] C. KARAKIZI, K. KARANTZALOS, M. VAKALOPOULOU, and G. ANTONIOU. “Detailed land cover mapping from multitemporal landsat-8 data of different cloud cover”. In: *Remote Sensing* 10.8 (2018), p. 1214.

[30] S. SHRESTHA and L. VANNESCHI. “Improved fully convolutional network with conditional random fields for building extraction”. In: *Remote Sensing* 10.7 (2018), p. 1135.

[31] E. MAGGIORI, Y. TARABALKA, G. CHARPIAT, and P. ALLIEZ. “Convolutional neural networks for large-scale remote-sensing image classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (2016), pp. 645–657.

[32] K. SIMONYAN and A. ZISSERMAN. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[33] S. JI, C. ZHANG, A. XU, Y. SHI, and Y. DUAN. “3D convolutional neural networks for crop classification with multi-temporal remote sensing images”. In: *Remote Sensing* 10.1 (2018), p. 75.

[34] B. BASSO, D. CAMMARANO, and E. CARFAGNA. “Review of crop yield forecasting methods and early warning systems”. In: *Proceedings of the first meeting of the scientific advisory committee of the global strategy to improve agricultural and rural statistics, FAO Headquarters, Rome, Italy*. 2013, pp. 18–19.

[35] A. K. PRASAD, L. CHAI, R. P. SINGH, and M. KAFATOS. “Crop yield estimation model for Iowa using remote sensing and surface parameters”. In: *International Journal of Applied Earth Observation and Geoinformation* 8.1 (2006), pp. 26–33.

[36] J. REN, Z. CHEN, Q. ZHOU, and H. TANG. “Regional yield estimation for winter wheat with MODIS-NDVI data in Shandong, China”. In: *International Journal of Applied Earth Observation and Geoinformation* 10.4 (2008), pp. 403–413.

[37] N. KIM and Y.-W. LEE. “Estimation of corn and soybeans yield using remote sensing and crop yield data in the United States”. In: *Remote Sensing for Agriculture, Ecosystems, and Hydrology XVI*. Vol. 9239. International Society for Optics and Photonics. 2014, 92390Y.

[38] D. JIANG, X YANG, N. CLINTON, and N. WANG. “An artificial neural network model for estimating crop yields using remotely sensed information”. In: *International Journal of Remote Sensing* 25.9 (2004), pp. 1723–1732.

[39] N. KIM and Y.-W. LEE. “Machine learning approaches to corn yield estimation using satellite images and climate data: a case of Iowa State”. In: *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography* 34.4 (2016), pp. 383–390.

[40] K. KUWATA and R. SHIBASAKI. “Estimating crop yields with deep learning and remotely sensed data”. In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2015, pp. 858–861.

[41] J. SUN, L. DI, Z. SUN, Y. SHEN, and Z. LAI. “County-Level Soybean Yield Prediction Using Deep CNN-LSTM Model”. In: *Sensors* 19.20 (2019), p. 4363.

[42] P. NEVAVUORI, N. NARRA, and T. LIPPING. “Crop yield prediction with deep convolutional neural networks”. In: *Computers and Electronics in Agriculture* 163 (2019), p. 104859.

[43] Q. YANG, L. SHI, J. HAN, Y. ZHA, and P. ZHU. “Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images”. In: *Field Crops Research* 235 (2019), pp. 142–153.

[44] V. VAPNIK. “Statistical Learning Theory Wiley-Interscience”. In: *New York* (1998).

[45] S. B. KOTSIANTIS, I. D. ZAHARAKIS, and P. E. PINTELAS. “Machine learning: a review of classification and combining techniques”. In: *Artificial Intelligence Review* 26.3 (2006), pp. 159–190.

[46] H. GAO, C. WANG, G. WANG, J. ZHU, Y. TANG, P. SHEN, and Z. ZHU. “A crop classification method integrating GF-3 PolSAR and Sentinel-2A optical data in the Dongting Lake Basin”. In: *Sensors* 18.9 (2018), p. 3139.

[47] H. DRUCKER, C. J. BURGES, L. KAUFMAN, A. J. SMOLA, and V. VAPNIK. “Support vector regression machines”. In: *Advances in neural information processing systems*. 1997, pp. 155–161.

[48] K. O’SHEA and R. NASH. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[49] A KAMILARIS and F. PRENAFETA-BOLDÚ. “A review of the use of convolutional neural networks in agriculture”. In: *The Journal of Agricultural Science* 156.3 (2018), pp. 312–322.

[50] J. SCHMIDHUBER. “Deep learning in neural networks: An overview”. In: *Neural networks* 61 (2015), pp. 85–117.

[51] G. E. DAHL, T. N. SAINATH, and G. E. HINTON. “Improving deep neural networks for LVCSR using rectified linear units and dropout”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 8609–8613.

[52] V. NAIR and G. E. HINTON. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[53] T. P. LILLICRAP, J. J. HUNT, A. PRITZEL, N. HEESS, T. EREZ, Y. TASSA, D. SILVER, and D. WIERSTRA. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[54] D. P. KINGMA and J. BA. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[55] D. WILSON and T. MARTINEZ. “The need for small learning rates on large problems”. In: vol. 1. Feb. 2001, pp. 115–119.

[56] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, and M. PALURI. “Learning spatiotemporal features with 3d convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[57] MOF. *Economic Survey, Fiscal Year 2009/10*. Tech. rep. Kathmandu, Nepal: Ministry of Finance (MOF), Government of Nepal, 2010.

[58] MOAC. *Statistical Information on Nepalese Agriculture, 2008/2009*. Tech. rep. Kathmandu, Nepal: Agri-business Promotion, Statistical Division, Ministry of Agriculture, and Cooperatives, 2009.

[59] S. GHIMIRE, S. M. DHUNGANA, V KRISHNA, N. TEUFEL, and D. SHERCHAN. *Biophysical and socio-economic characterization of cereal production systems of Central Nepal*. CIMMYT, 2013.

[60] ESA. *Sentinel-2 User Handbook, Issue 1, Rev 2*. Tech. rep. ESA Standard Document. European Space Agency, 2015.

[61] F. GASCON, E. CADAU, O. COLIN, B. HOERSCH, C. ISOLA, B. L. FERNÁNDEZ, and P. MARTIMORT. "Copernicus Sentinel-2 mission: products, algorithms and Cal/Val". In: *Earth Observing Systems XIX*. Vol. 9218. International Society for Optics and Photonics. 2014, 92181E.

[62] R RICHTER and D SCHLÄPFER. "Atmospheric/Topographic Correction for Satellite Imagery (ATCOR-2/3 User Guide, Version 8.3. 1, February 2014)". In: *ReSe Applications Schläpfer, Langeggweg* 3 (2013).

[63] F. VUOLO, M. ŻÓŁTAK, C. PIPITONE, L. ZAPPA, H. WENNG, M. IMMITZER, M. WEISS, F. BARET, and C. ATZBERGER. "Data service platform for Sentinel-2 surface reflectance and value-added products: System use and examples". In: *Remote Sensing* 8.11 (2016), p. 938.

[64] R. KARKI, R. TALCHABHADEL, J. AALTO, and S. K. BAIDYA. "New climatic classification of Nepal". In: *Theoretical and applied climatology* 125.3-4 (2016), pp. 799–808.

[65] S. SHRESTHA and T. BAIDAR. "Spatial Distribution and Temporal Change of Extreme Precipitation Events on the Koshi Basin of Nepal". In: *Nepalese Journal of Geoinformatics* 17.1 (2018), pp. 38–46.

[66] H. SONG, Y. KIM, and Y. KIM. "A patch-based light convolutional neural network for land-cover mapping using Landsat-8 images". In: *Remote Sensing* 11.2 (2019), p. 114.

[67] I. GOODFELLOW, Y. BENGIO, and A. COURVILLE. *Deep learning*. MIT press, 2016.

[68] T. TAKASE, S. OYAMA, and M. KURIHARA. "Why does large batch training result in poor generalization? A comprehensive explanation and a better strategy from the viewpoint of stochastic optimization". In: *Neural computation* 30.7 (2018), pp. 2005–2023.

Masters Program in **Geospatial Technologies**

RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA: A CASE STUDY OF TERAI DISTRICTS OF NEPAL

Tina Baidar

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

2020

RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA:
A CASE STUDY OF TERAI DISTRICTS OF NEPAL

Tina Baidar

Masters
Program
in **Geospatial**
Technologies

Supported by:

Education and Culture

ERASMUS MUNDUS