Crop Yield Forecasting

M.Tech Dissertation-1 2021-22 Project Report

Submitted by

Mr. Souvik Roy
20-27-08

Under the kind guidance of

Mr. Manmeet Singh (Scientist 'C', IITM Pune)
Dr. Ramkrishna Bharath (Asst. Professor, Department of Applied Mathematics, DIAT, Pune)

Department of Applied Mathematics
Defence Institute of Advanced Technology
Girinagar, Next to Khadakwasla Dam, Pune, Maharashtra-411025
(Affiliated to Ministry of Defence)
Dec 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective

Agriculture has always been an important sector for every person who is involved directly or indirectly in this sector. The traditional methods are still widely used in agriculture but as the advancement of technology is increasing day by day so the effectiveness of products produced by agriculture such as crops, fruits, and many more products are getting enhanced. The help of modern technologies such as Artificial Intelligence, Remote Sensing, and various other technologies are helping the agriculture sector to boom faster which is a great sign for people involved in this sector. The main goal of this project is to use one of the advancements which are using Deep Learning model with the help of UNET architecture which will be used to forecast the yield of vegetation or crops for which in this project is NDVI (Normalized Difference Vegetation Index) and by the help of Meteorological methods to convert the data of vegetation or crops into a suitable format to make it useful for the Deep Learning model to proper forecast the yields of vegetation.

1.2 Motivation

The vegetation always plays a vital role in the enhancement of ecosystems but in recent times due to heavy involvement of human interaction with nature such as exploitation of forests, illegal mining and many more have played a huge role in the declining of the vegetation and the related resources like food production, crop yielding, etc. Normalized Difference Vegetation Index (NDVI) is extensively used in most of the research works connected to vegetation. Several methods had been used in the last few years related to NDVI research works such as ANN (Artificial Neural Networks), Multiple Layer Perceptron (MLP), and many more deep learning architectures at the beginning phase of these research works but have some limitations which is the reason extensive studies are made to use other deep learning models for NDVI. The main purpose of the project is that it will use a newly developed concept or model named Cubed Sphere [2]. The data is for the whole world and the format of the NDVI will be converted into the Cubed Sphere format and then these files will be further applied to the Deep Learning model which is UNET architecture [1]. After the model is applied then evaluation metrics will be used to test this newly made concept of Cubed Sphere is working for NDVI data.

Chapter 2

Literature Survey

In the year 2021, Manmeet Singh and Bipin Kumar et.al. [1], suggested that the understanding of the relationship of other variables with the formation of precipitation is very beneficial. The UNET architecture is used to learn the global data-driven models for precipitation and the results showed that the residual learning-based UNET architecture can explain the relationships for the precipitation variable and it can be helpful in the dynamical operational models which are compared with the results that further improve the forecasting of precipitation.

Jonathan A.Weyn and Dale R.Durran et.al. [2], proposed further improvements in CNN (Convolutional Neural Networks), created an offline volume conservative mapping to the newly formed concept called Cubed Sphere, and also minimized the loss function that was occurring in the various steps of the prediction sequence. The model predicts weather forecasts which are stable and does so for several weeks and also uses few input atmospheric state variables.

Chapter 3

Description of Data Variables

Vegetation data from the satellite display the thickness of growth of plants across Earth. The most widely used vegetation index is called the Normalized Difference Vegetation Index (NDVI) which ranges from -1 to 1. The negative values show water and low values as 0.1 correlate with areas having rock, sand, etc. The values from 0.2 to 0.3 indicate shrub and grassland while values from 0.6 to 0.8 represent rainforests.

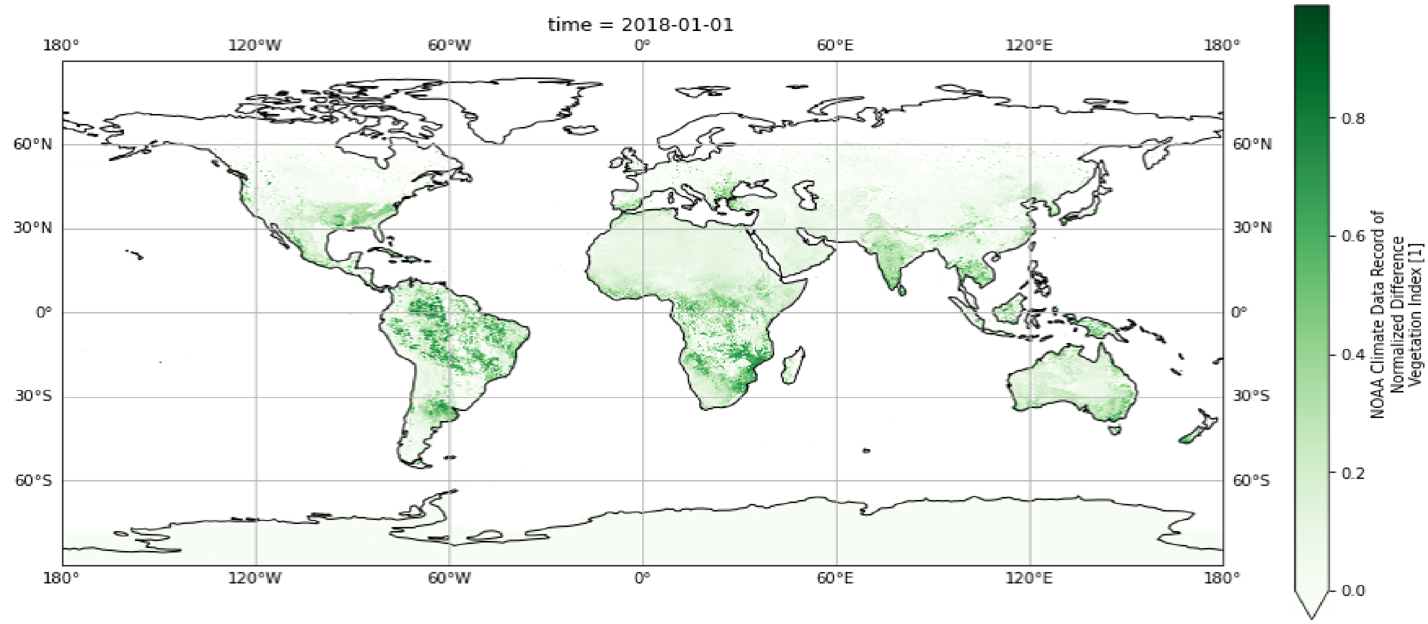

Figure 3.1: Plot of NDVI spread over the entire globe for 2018-01-01

The NDVI data is in NETCDF format (Network Common Data Form) which is used to store data variables generally of climate area such as temperature, humidity, etc. To read these files in Python there is a library named "Xarray" which has several inbuilt commands which handle NETCDF files.

S.No	Target Variable	Name	Resolution	Units	Min	Max
1.	NDVI	Normalized Difference Vegetation Index	3600*7200	1	-0.0999	0.9307

Table 3.1: Details of NDVI Variable

Six input variables will be fed into the deep learning model along with the target variable NDVI. The input variables that are being used are taken to get correct mapping with the target variable like precipitation, wind, etc. Here the input variables uas and vas is used as a parameter for a wind variable which is the square root of the square of uas plus the square of the vas.

S.No	Input Variables	Name	Resolution	Units
1	hfls	Surface Upward Latent Heat Flux	143*144	W m-2
2	hfss	Surface Upward Sensible Heat Flux	143*144	W m-2
3	mrsos	Moisture in Upper Portion of Soil Column	143*144	kg m-2
4	uas	Eastward Near-Surface Wind	143*144	m s-1
5	vas	Northward Near-Surface Wind	143*144	m s-1
6	pr	Precipitation	143*144	Kg m-2 s-1
7	tas	Near-Surface Air Temperature	143*144	K

Table 3.2: Details of the Input Variables

Chapter 4

Methodology

The standard representation of the model with a detailed explanation is explained below. The block diagram provides a clear explanation of each of the steps used to generate the Cubed Sphere before using it for the deep learning model.

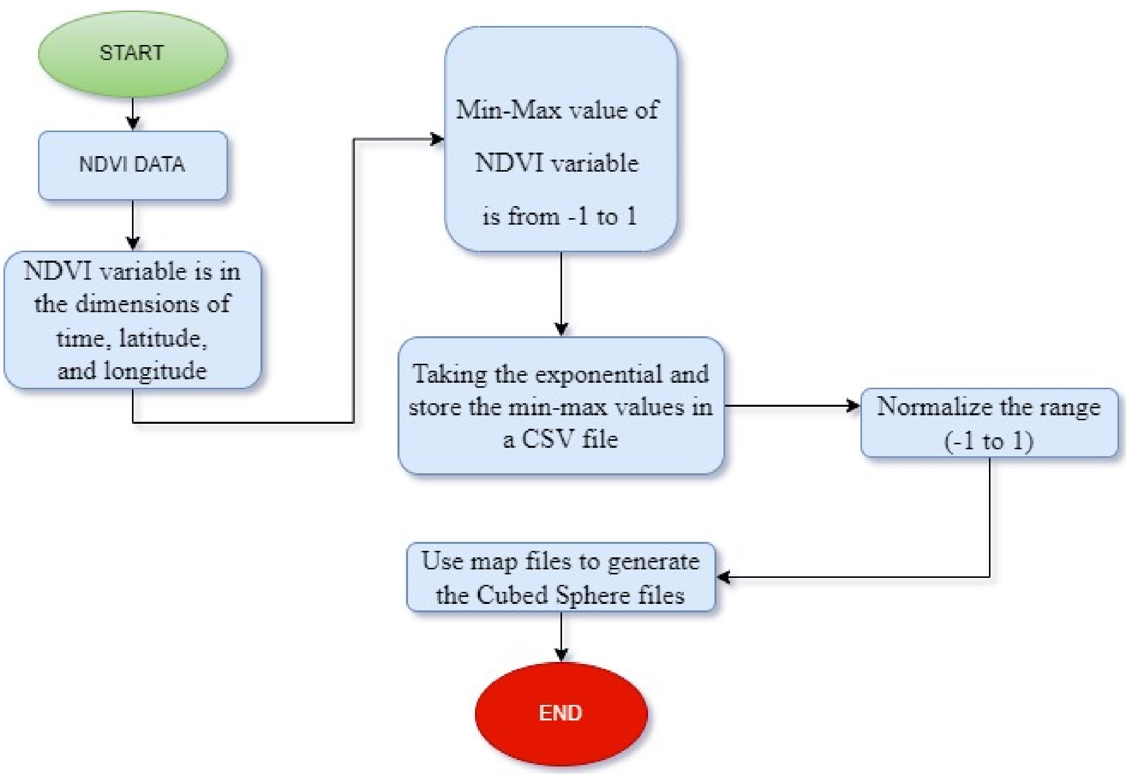

Figure 4.1: Block diagram for generating Cubed Sphere files

In Fig. 4.1, NDVI data is downloaded which is from the year 1981 to 2021. The data downloaded is the daily data which means the NDVI data is available every day. There are two years 1981 and 2021 where there are missing data for several days so these two years have been discarded for better performance of the Deep Learning model. The next process is to recognize if a variable is in tripolar grid or not so which in this case is no as the variables are in normal time latitude and longitude format which is required for further processing. Since NDVI is a target variable and the input variables used for the NDVI variable are in normal time, latitude, and longitude format so the target variable also needs to match that format which is already available for NDVI. If there were tripolar grid format then CDO (Climate Data Operator) will be used to convert into general time, latitude, and longitude format and use it for further process. Then there is a need to merge the daily data into a single file for every year and finally need to mask the land as a batch of 10 years files area. Due to the mask of the land, it becomes essential to mask the land and oceanic variables needed for the exponential of the data array. The NDVI variable is a land variable. The resulting data array's value range becomes so huge after taking exponential that there is a need to normalize to the range from 0 to 1 using min-max normalization and the maximum and minimum values are stored in a CSV file. After all these steps there is a reduction of the value range from -1 to 1. For NDVI the minimum and maximum values are already in the range -1 to 1 so there is no need for min-max normalization and can take the exponential directly. The last step is to use the Map file generation and Cubed Sphere generation that uses DLWP (Deep Learning Weather Prediction) which in the background uses TempestRemap [2]. The map file that is generated at first is stored in a location that can be reused to save computation time as it is quite expensive. After the map file is generated then the ApplyOfflineMap executable will generate a remapped temporary file that helps, in turn, to give results in the Cubed Sphere generation with CubeSphereRemap() module imported from DLWP.remap. After this, the NDVI Cubed Sphere files are used in UNET architecture and evaluation metrics are used to check the performance of the model.

Chapter 5

Description of Deep Learning Model Used

The flowchart illustrates the deep learning model used, variables given as input and target, and the metrics used to evaluate the model.

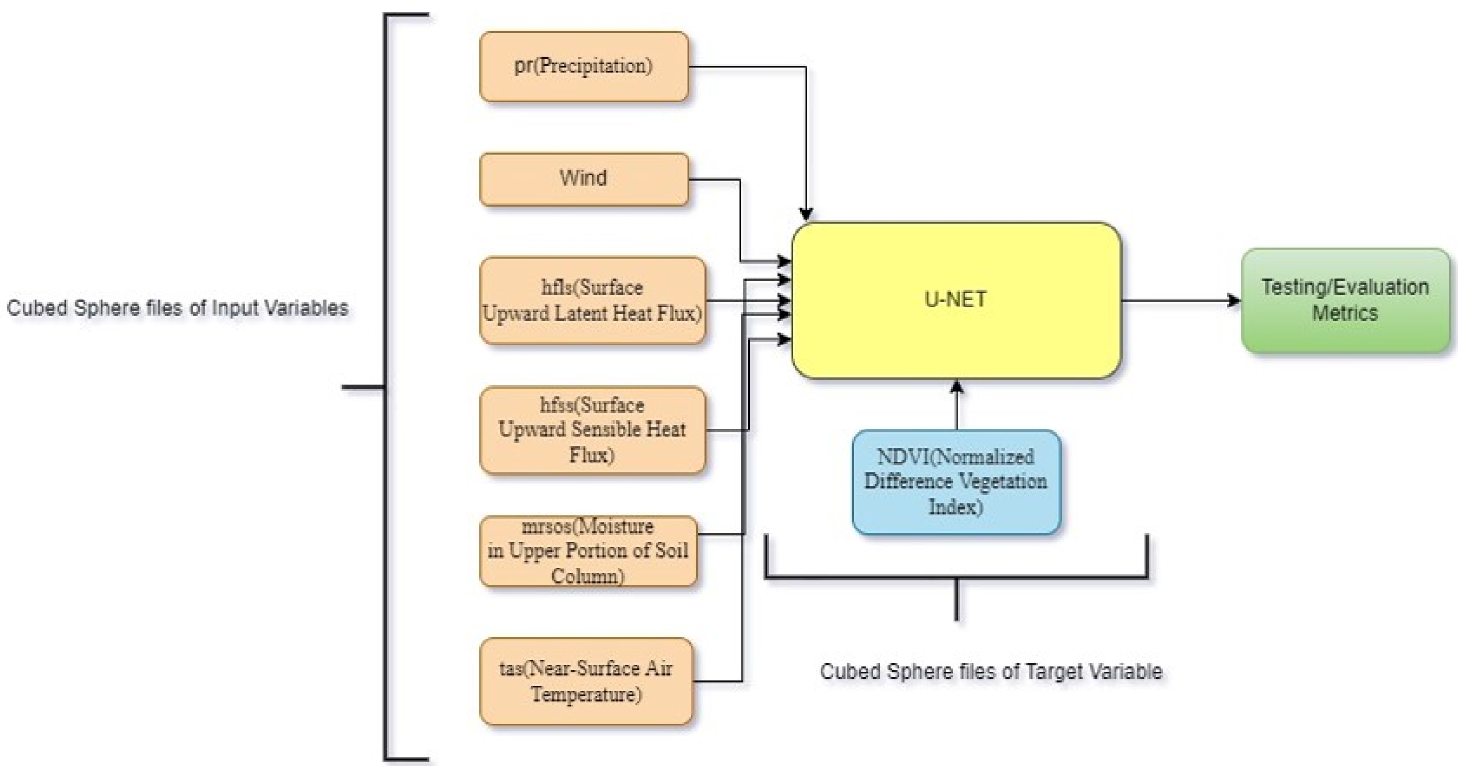

Figure 5.1: Illustration of working of Deep Learning Model for NDVI

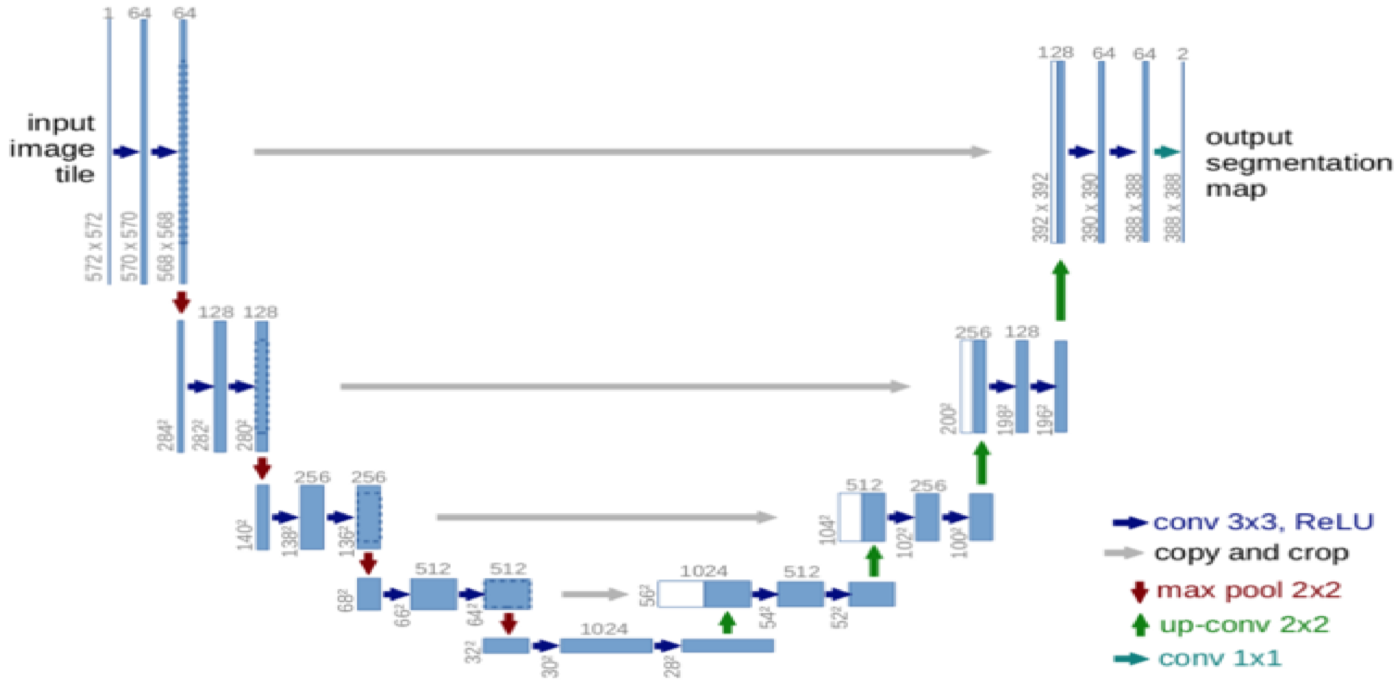

Figure 5.2: Schematic for UNET architecture

The Cubed Sphere files of both the input and target variables are fed into the UNet model. In the figure, the architecture forms like a U shape structure. It shows symmetricity nature and involves two parts which are left part is named as the contracting path that constitutes the convolutional methods and the right part is named as the expansive path that constitutes by transposed 2D Convolutional Layers.

Chapter 6

Results

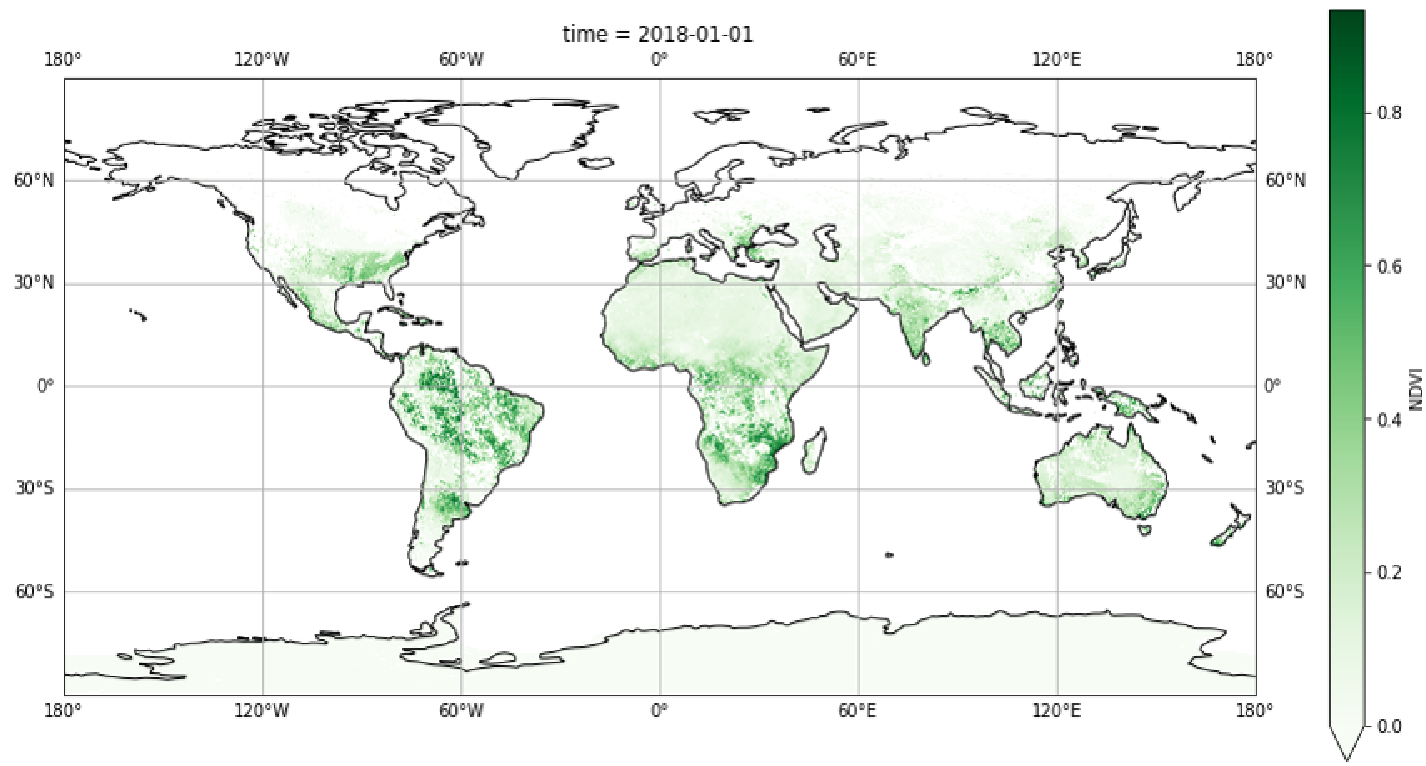

Figure 6.1: Plot of the NDVI spread from ten-ten year batch files for 2018-01-01

```
<xarray.Dataset>
Dimensions:  (lat: 720, lon: 1440, time: 3651)
Coordinates:
  * time     (time) datetime64[ns] 2011-01-01 2011-01-02 ... 2020-12-31
  * lon      (lon) float32 0.125 0.375 0.625 0.875 ... 359.1 359.4 359.6 359.9
  * lat      (lat) float32 -89.88 -89.62 -89.38 -89.12 ... 89.38 89.62 89.88
Data variables:
    NDVI     (time, lat, lon) float64 ...
Attributes:
    CDI:          Climate Data Interface version 1.9.10 (https://mpimet.mpg.d...
    Conventions:  CF-1.6
    history:      Mon Sep 27 21:38:22 2021: cdo remapbil,/lus/dal/mtechstuden...
    CDO:          Climate Data Operators version 1.9.10 (https://mpimet.mpg.d...
```

Figure 6.2: Details of one of the ten-ten year batch files

The above two figures show the NDVI spread over the whole globe for 2018-01-01 and also the details of one of the ten-ten year batch files.

```
remap_converted_grid_NDVI_1982_1991_normalized_CS96.nc  remap_converted_grid_NDVI_1997_2006_normalized_CS96.nc
remap_converted_grid_NDVI_1983_1992_normalized_CS96.nc  remap_converted_grid_NDVI_1998_2007_normalized_CS96.nc
remap_converted_grid_NDVI_1984_1993_normalized_CS96.nc  remap_converted_grid_NDVI_1999_2008_normalized_CS96.nc
remap_converted_grid_NDVI_1985_1994_normalized_CS96.nc  remap_converted_grid_NDVI_2000_2009_normalized_CS96.nc
remap_converted_grid_NDVI_1986_1995_normalized_CS96.nc  remap_converted_grid_NDVI_2001_2010_normalized_CS96.nc
remap_converted_grid_NDVI_1987_1996_normalized_CS96.nc  remap_converted_grid_NDVI_2002_2011_normalized_CS96.nc
remap_converted_grid_NDVI_1988_1997_normalized_CS96.nc  remap_converted_grid_NDVI_2003_2012_normalized_CS96.nc
remap_converted_grid_NDVI_1989_1998_normalized_CS96.nc  remap_converted_grid_NDVI_2004_2013_normalized_CS96.nc
remap_converted_grid_NDVI_1990_1999_normalized_CS96.nc  remap_converted_grid_NDVI_2005_2014_normalized_CS96.nc
remap_converted_grid_NDVI_1991_2000_normalized_CS96.nc  remap_converted_grid_NDVI_2006_2015_normalized_CS96.nc
remap_converted_grid_NDVI_1992_2001_normalized_CS96.nc  remap_converted_grid_NDVI_2007_2016_normalized_CS96.nc
remap_converted_grid_NDVI_1993_2002_normalized_CS96.nc  remap_converted_grid_NDVI_2008_2017_normalized_CS96.nc
remap_converted_grid_NDVI_1994_2003_normalized_CS96.nc  remap_converted_grid_NDVI_2009_2018_normalized_CS96.nc
remap_converted_grid_NDVI_1995_2004_normalized_CS96.nc  remap_converted_grid_NDVI_2010_2019_normalized_CS96.nc
remap_converted_grid_NDVI_1996_2005_normalized_CS96.nc  remap_converted_grid_NDVI_2011_2020_normalized_CS96.nc
```

Figure 6.3: Cubed Sphere files generated for NDVI

```
<xarray.Dataset>
Dimensions:  (face: 6, height: 96, time: 3651, width: 96)
Coordinates:
  * time     (time) datetime64[ns] 2011-01-01 2011-01-02 ... 2020-12-31
  * face     (face) int64 0 1 2 3 4 5
  * height   (height) int64 0 1 2 3 4 5 6 7 8 9 ... 87 88 89 90 91 92 93 94 95
  * width    (width) int64 0 1 2 3 4 5 6 7 8 9 ... 86 87 88 89 90 91 92 93 94 95
Data variables:
    lon      (face, height, width) float64 ...
    lat      (face, height, width) float64 ...
    NDVI     (time, face, height, width) float32 ...
```

Figure 6.4: Details of one of Cubed Sphere files

The above two figures show the cubed sphere files generated and with details of one cubed sphere file.

```
457/457 [==============================] - 42s 93ms/step - loss: 0.0082 - val_loss: 0.0093
Epoch 00009: early stopping
Training for years  2006   2015
Loading data into RAM
Epoch 1/1000
457/457 [==============================] - 42s 92ms/step - loss: 0.0083 - val_loss: 0.0091
Epoch 2/1000
457/457 [==============================] - 43s 94ms/step - loss: 0.0083 - val_loss: 0.0098
Epoch 3/1000
457/457 [==============================] - 42s 92ms/step - loss: 0.0082 - val_loss: 0.0092
Epoch 00003: early stopping

Loss =  [[0.032542359083890915, 0.022660406306385994, 0.02017412707209587, 0.01880934089422226, 0.017880450934171677, 0.01720198057591915, 0.
01679106429219246, 0.016419624909758568, 0.015891429036855698], [0.015268132090568542, 0.01490691676735878, 0.014597110450267792, 0.014330551
028251648, 0.014108269475400448], [0.01403275690972805, 0.013781457208096981, 0.013702556490898132, 0.013470248319208622, 0.01340556796640157
7, 0.013161131180822085, 0.012984246015548706], [0.015553186647593975, 0.015187819488346577, 0.015062198042869568, 0.014854319393634796], [0.0
14528445899486542, 0.014299586415290833, 0.014213772490620613, 0.014077205210924149, 0.013926023617386818, 0.013855796307325363], [0.01435413
770377636, 0.014363899827003479, 0.014075792394578457, 0.013938622549176216], [0.014036154374480247, 0.013864110223948956, 0.0137669267132878
3, 0.013492237776517868, 0.013439773581922054], [0.01314842514693737, 0.012917358428239822, 0.012763662263751003, 0.012632202357053757], [0.01
271376945078373, 0.012471224181354046, 0.012325670570135117], [0.012836641632020473, 0.012513389810919762, 0.012483302503824234, 0.0123085612
43116856, 0.012223646976053715], [0.012818314135074615, 0.012602569535374641, 0.01244228146970272], [0.012908319011330605, 0.0126351192593574
52, 0.012443402782082558, 0.012377987615764141], [0.012796362861990929, 0.012568412348628044, 0.012421134859323502], [0.009971885941922665, 0
.00980697013437748, 0.009711808525025845, 0.009663485921919346, 0.009605466388816595, 0.009536658413708821, 0.009519926272332668, 0.00945346802
4730682], [0.009952455759048462, 0.009803271852433681, 0.009777650237083435, 0.009685602970421314, 0.009615735150873661], [0.0095157474279403
69, 0.009430693462491035, 0.009342236444354057, 0.009301521815359592], [0.009614360518753529, 0.009459506720304449, 0.009445018135011196, 0.00
9359639137983322], [0.009644316509366035, 0.009598265402019024, 0.009493908844888821, 0.009456753730773926], [0.00941410567611456, 0.009301303
885877132, 0.009252762421965599, 0.009223299100995064], [0.009115212596952915, 0.009059499017894268, 0.008975263684988022, 0.0089547624811530
11, 0.008879823610186577, 0.008835673332214355], [0.009040392935276031, 0.008889828808605671, 0.008863287977874279, 0.008851510472595692], [0
.008940340951085509, 0.008850873447954655, 0.008848713710904121, 0.008745425380766392, 0.008741197176277637, 0.008702361956238747, 0.008712687
529623508], [0.008872518315911293, 0.008819853886961937, 0.008700556121766567], [0.008555494248867035, 0.008488431572914124, 0.00843283720314
5027, 0.008383524604141712, 0.008370394818484783, 0.008326745592057705, 0.008344366215169 43, 0.008282762952148914, 0.008246351964771748], [0.
008287818171083927, 0.008273148909211159, 0.008157915435731411]]
```

Figure 6.5: Epochs and Losses calculated for NDVI variable after one iteration

The training set is taken from 1982 to 2015, the validation set is taken for two years which are 2016 and 2017 and the test set is taken from 2018 to 2020.

Bibliography

[1] Manmeet Singh, Bipin Kumar, Suryachandra Rao, Sukhpal Singh Gill, Rajib Chattopadhyay, Ravi S Nanjundiah, Dev Niyogi. Deep learning for improved global precipitation in numerical weather prediction systems. arXiv.org, Physics, Atmospheric and Oceanic Physics (physics.ao-ph); Machine Learning (cs.LG), arXiv:2106.12045v2 [physics.ao-ph], 24 Aug 2021.

[2] Jonathan A.Weyn, Dale R.Durran, Rich Caruana. Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere. arXiv:2003.11927v1 [physics.ao-ph], 15 Mar 2020.

[3] Xiao Xiang Zhu and (2017). Deep learning in remote sensing: a review. CoRR, abs/1710.03959.

[4] Lopresti, M. F., Di Bella, C. M., Degioanni, A. J. (2015). Relationship between MODIS-NDVI data and wheat yield: A case study in Northern Buenos Aires province, Argentina. Information Processing in Agriculture, 2(2), 73–84. doi:10.1016/j.inpa.2015.06.001.

[5] Huang, J., Wang, X., Li, X., Tian, H., Pan, Z., Prasad, A., … Cutler, M. (2013). Remotely Sensed Rice Yield Prediction Using MultiTemporal NDVI Data Derived from NOAA's-AVHRR. PLoS ONE, 8(8), e70816. doi:10.1371/journal.pone.0070816