# Design and development of Nodel Deep Learning algorithm Hyperspectral Image Classification

## BTP 2024

Presented by : Priyanshu Gupta - 112001033

Mentor : Dr. Satyajit Das

## Problem statement

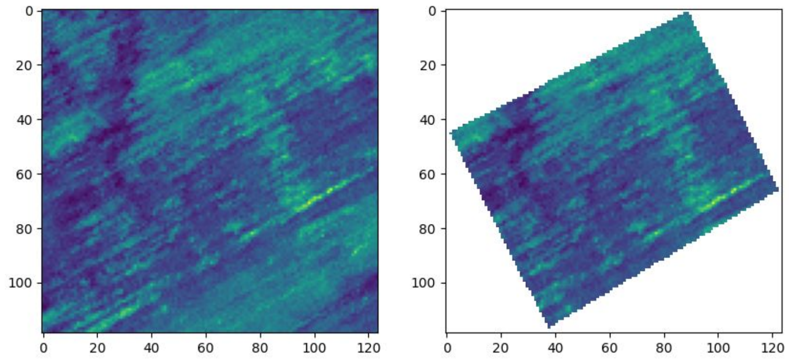
- Designing a nodal deep learning algorithm for prediction of soil nutrients present in the agricultural soil

- Deployment of the algorithms to test them on real satellite data.

## Dataset 1

- The dataset comprises 1732 hyperspectral images each encompassing 150 wavelength bands within the range of 462 to 492 nm having a spectral resolution of 3.2 nm.
- The training dataset includes information on four essential soil nutrients, namely Phosphorus (P), Potassium (K), Magnesium (Mg), and soil pH.
- Non-uniform distribution of image dimensions across the dataset ranging from 11 x 11 to 250 x 250.

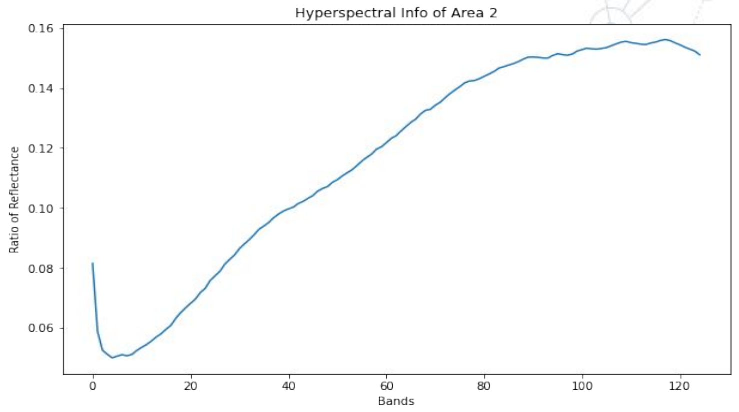
Representation of band 1 (462.08 nm)



| Sample index | P | K | Mg | pH |
|---|---|---|---|---|
| 1731 | 29.5 | 146 | 133 | 6.3 |

# Dataset 2

- The dataset comprises 679 rows of hyperspectral data, each encompassing 125 wavelength bands within the range of 454 to 950 nm with a spectral resolution of 4 nm.
- This dataset was captured on the same patch of land in a five-day field campaign in May 2017 in Karlsruhe, Germany.
- An undisturbed soil sample is the centerpiece of the measurement setup. The soil sample consists of bare soil without any vegetation and was taken in the area near Waldbronn, Germany.



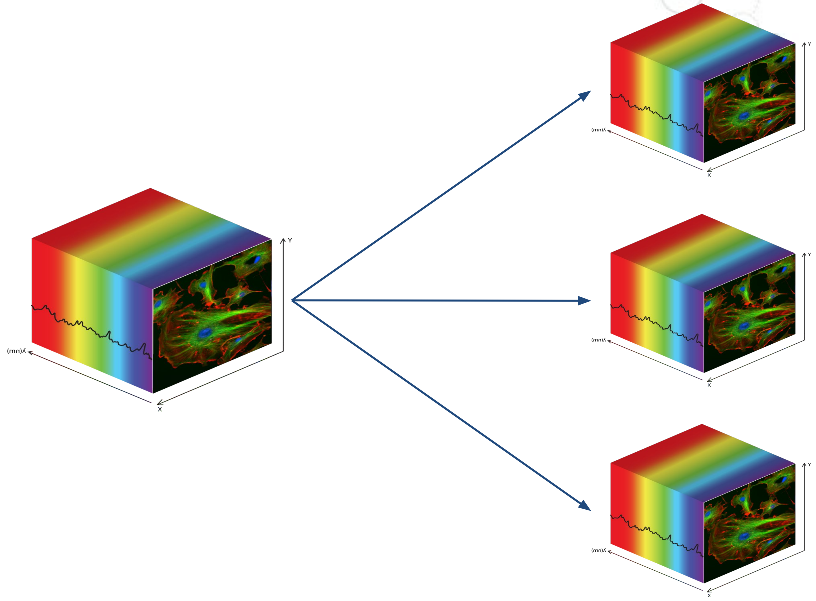
| sample_index | Soil Moisture | Soil Temperature |
|---|---|---|
| 2 | 33.9 | 36.04 |

# Data Augmentation (Phase 1)

- In the first half, algorithm tries to find 11x11 patch without masked values and if found, adds a noise to the data and add in augmented samples.

- If no such patch is found, we take the biggest square patch instead.

- We perform the step 1 ten times finding random such patch and if we don't find such patch we proceed to step 2.



# Preprocessing and Models (Phase 1)

- Training data conversion to : arr : Spectral Curve, dXdl, d2Xdl2, d3Xdl3, dXds1 : Next order Derivatives, s0, s1, s2, s3, s4: SVD, real, imag, reals, imags : FFT, cAw1, cAw2 : Wavelets

- Random forest and XGBoost are trained on 100 estimators and sequential model used has 6 dense layers with 6 dropout layers to reduce overfitting.

| Model | RMSE |
|---|---|
| Random Forest | 46.83 |
| XGBoost | 46.14 |
| SVR | 54.18 |
| Deep Learning (Sequential) | 38.54 |

# 3D and 2D CNN : Preprocessing 1

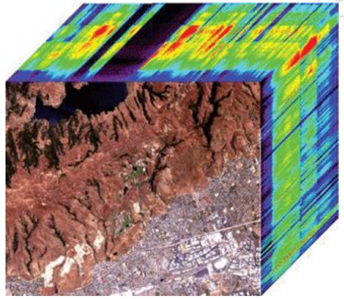
- The resizing process using OpenCV (cv2) involves the multiplication of the masked values and data matrix, applying inter area interpolation to **adjust the image size to 11 x 11**.

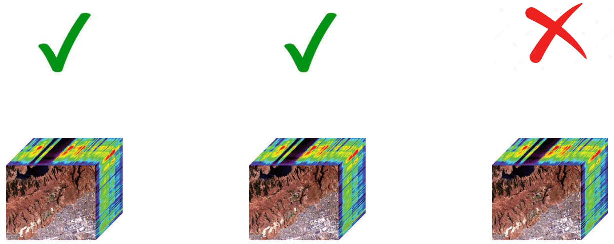
- During this operation, the **mask receives a weight of 0**, effectively disregarding its influence on the final result, while the **actual data is assigned a weight of 1**, ensuring its preservation.

- This approach maintains the integrity of the real data while accommodating the resizing requirements, effectively handling masked regions with minimal impact on the overall transformation outcome.

## 3D and 2D CNN : Preprocessing 2

- At first, the smallest patch was of size 11 x 11, which constituted around 650 images. So for the selection of a consistent image size, an 11 x 11 size was chosen.
- Next, for the selection of the threshold number of unmasked pixels for consideration of the patch for training,we looked into the smallest images and found the minimum number of unmasked pixels out of 121.
- We divided each patch into 11 x 11 multiple areas, and whichever area had more unmasked pixels as compared to the threshold was taken into consideration





# 3D CNN and 2D CNN

- Moving to 3D, as we have reflectance values for each pixel corresponding to a band and image size to be the same, the input was given a shape of 11 x 11 x 150.
- Incorporates 3D convolutional layers with **32 and 64 filters**, and a (3, 3, 3) kernel size, employing ReLU activation functions for feature extraction.Applies average pooling layers with a (2, 2, 2) pool size after each convolutional layer.
- Includes **two fully connected dense layers with 128 and 4 neurons** respectively, activated using ReLU and linear functions, for regression tasks.
- As the images were in 3D, the 150 wavelength bands were taken as channels in the case of **2D CNN**, and image **size was fixed to be 11 x 11**.
- Includes two convolutional layers with same filters and a (3, 3) kernel size using dense layers with 64 and 4 neurons.

## HybridSN 3D CNN and 3D-2D CNN

- Sequentially integrates 3D convolutional layers with varying filter sizes (8, 16, 32, 64) and kernel sizes ((3, 3, 7), (3, 3, 5), (3, 3, 3), (3, 3, 3)) for hierarchical feature extraction.

- Incorporates fully connected dense layers with 256 and 128 neurons respectively, utilizing ReLU activation functions and dropout regularization (40% dropout rate) to mitigate overfitting.

- In 3D-2D CNN we utilize a combination of 3D convolutional layers followed by a reshaping operation to accommodate subsequent 2D convolutional layers and rest same followed by dense layer with 4 neurons.

## 3D and 2D CNN : Result

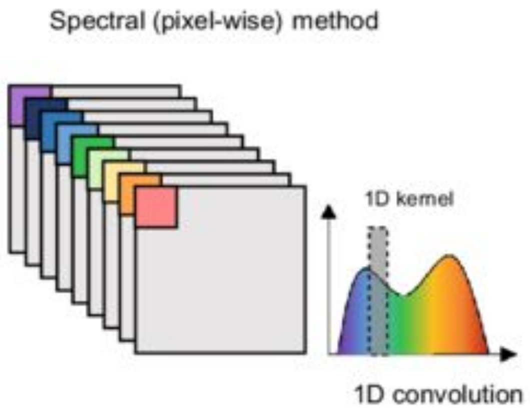
| | 2D CNN | | 3D CNN | | HybridSN 3D CNN | | HybridSN 3D-2D CNN | |
|---|---|---|---|---|---|---|---|---|
| | *RMSE* | *MAE* | *RMSE* | *MAE* | *RMSE* | *MAE* | *RMSE* | *MAE* |
| **Preprocessing 1** | 34.66 | 21.97 | 34.02 | 22.3 | 33.68 | 20.9 | 36.6 | 21.38 |
| **Preprocessing 2** | 35.88 | 22.78 | 41.43 | 27.24 | 56.4 | 30.16 | 38.24 | 22.67 |

# 1D CNN : Preprocessing

- Based on the results, it appears that the correlation among the pixels is not good enough. So, thought comes to finding correlation between bands.

- We iterate over all the patches in the images. For each unmasked pixel within a patch, we extract a 1D vector of length 150, representing the spectral values across bands. Train test split was made 80-20 for each patch.

- In accordance with the prior work, we also checked the results by adding first order and second order derivatives along with the 150 vector.



## 1D CNN : LucasCNN & HuEtAl

- **LucasCNN :** Features four convolutional layers with 32 and 64 filters, followed by max-pooling layers.
- Output is flattened and processed through two fully connected layers with 120 and 160 neurons. Final layer has four neurons with linear activation, suitable for regression tasks.

- **HuEtAl :** 1D CNN architecture proposed by Wei Hu et al. (2014).
- Consists of a single convolutional layer with 20 filters and tanh activation.
- Output is flattened and connected to a dense layer with 100 neurons, followed by a final output layer with four neurons for regression.

## 1D CNN : LiuEtAl & Lucas CoordConv

- **LiuEtAl :** 1D CNN architecture proposed by Lanfa Liu et al. (2018).
- Consists of four convolutional layers with 32 and 64 filters, followed by max-pooling.
- Flattened output is connected to a dense layer with four neurons, suitable for regression.

- **Lucas CoordConv :** Variant of LucasCNN incorporating CoordConv layers.
- Employs four convolutional layers with 32, 64, and 128 filters. Uses CoordConv layers in the first convolutional layer to incorporate coordinate information.
- Flattened output processed through two fully connected layers before the final regression output layer.

- **Simple CNN :** Comprises two convolutional layers with 64 filters and a kernel size of 3, followed by max-pooling.
- Flattened output is passed through a dense layer with 64 neurons and ReLU activation with output layer of 4 neurons.

# 1D CNN : Result

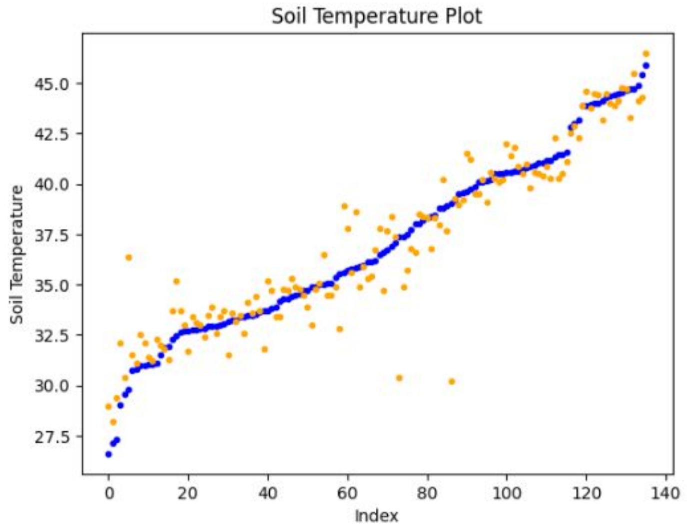
| | LucasCNN | | HuEtAl | | LiuEtAl | | Lucas Coordconv | |
|---|---|---|---|---|---|---|---|---|
| | *RMSE* | *MAE* | *RMSE* | *MAE* | *RMSE* | *MAE* | *RMSE* | *MAE* |
| **1d normalised (150)** | 32.3 | 18.8 | 37.2 | 22.6 | 34.14 | 20.35 | 32.5 | 19.01 |
| **1d normalised (450)** | 32.09 | 18.9 | 33.88 | 20.31 | 33.16 | 19.69 | 31.55 | 18.27 |

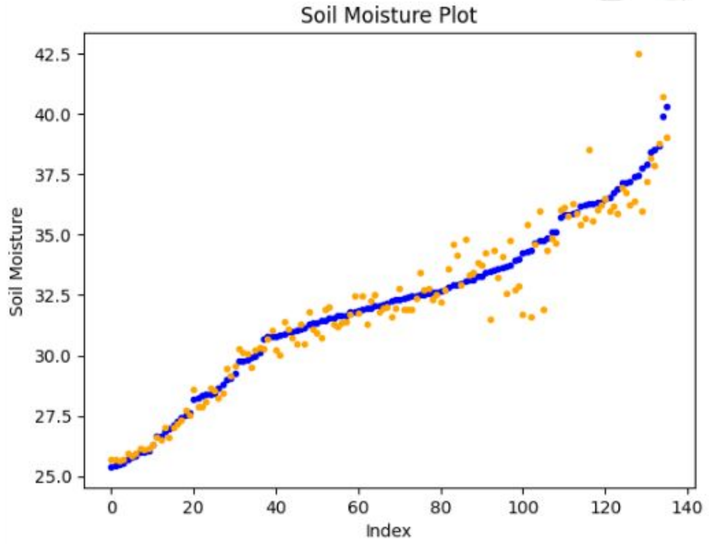
## Dataset 2 : Results

- Having the data to be in the format of 125 reflectance value vectors for each reading, our work of preprocessing for applying 1D CNN was relieved.

- After normalization, applying all 5 models to the dataset yielded the results as follows :

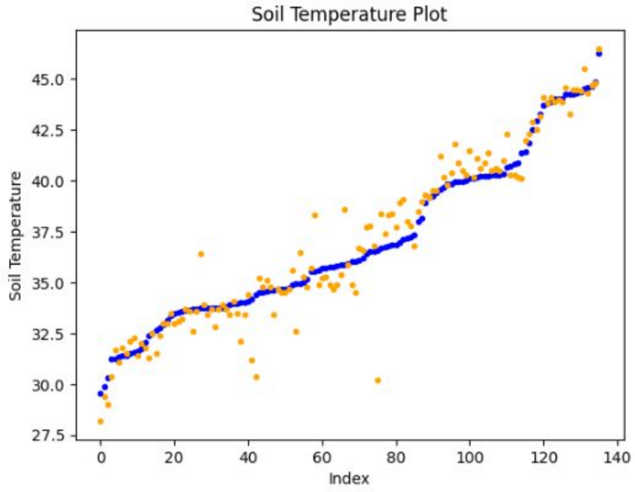
| **LucasCNN** | | **HuEtAl** | | **LiuEtAl** | | **Lucas coord conv** | | **Simple cnn** | | **PCA components** |
|---|---|---|---|---|---|---|---|---|---|---|
| ***MSE*** | ***MAE*** | ***MSE*** | ***MAE*** | ***MSE*** | ***MAE*** | ***MSE*** | ***MAE*** | ***MSE*** | ***MAE*** | |
| 3.4 | 1.31 | 0.98 | 0.68 | 2.1 | 1.1 | 1.49 | 0.85 | 0.85 | 0.82 | **normal** |
| 7.13 | 1.91 | 2.82 | 1.18 | 4.75 | 1.75 | 6.72 | 2.1 | 2.79 | 1.29 | **pca 2** |
| 2.43 | 1.15 | 2.11 | 1.03 | 6.88 | 2.24 | 2.19 | 1.1 | 1.66 | 0.89 | **pca4** |
| 3.11 | 1.48 | 1.13 | 0.7 | 3.9 | 1.5 | 1.16 | 0.78 | 1.63 | 0.96 | **pca 6** |
| 1.37 | 0.82 | 1.05 | 0.67 | 3.04 | 1.27 | 1.08 | 0.66 | 1.05 | 0.65 | **pca 8** |

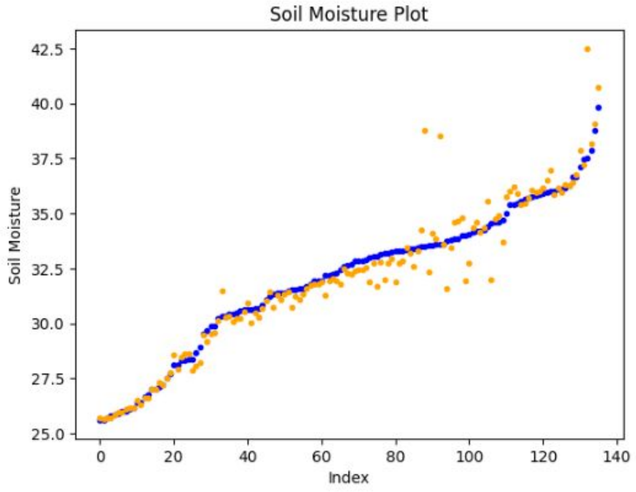

Soil Temperature Plot
Soil Temperature
Index
27.5
30.0
32.5
35.0
37.5
40.0
42.5
45.0
0
20
40
60
80
100
120
140


Soil Moisture Plot
Soil Moisture
Index
25.0
27.5
30.0
32.5
35.0
37.5
40.0
42.5
0
20
40
60
80
100
120
140

## Random Forest

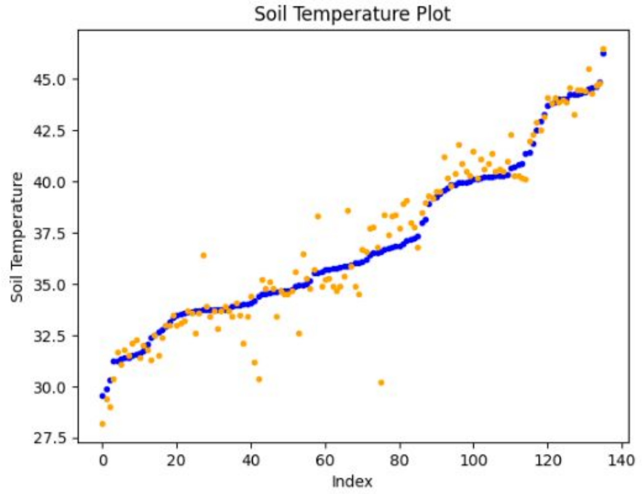
- Using the random forest regressor for training on the data using 200 estimators gave us mse 1.06 and mean absolute error to be 0.66 and moreover the plots are as follows.

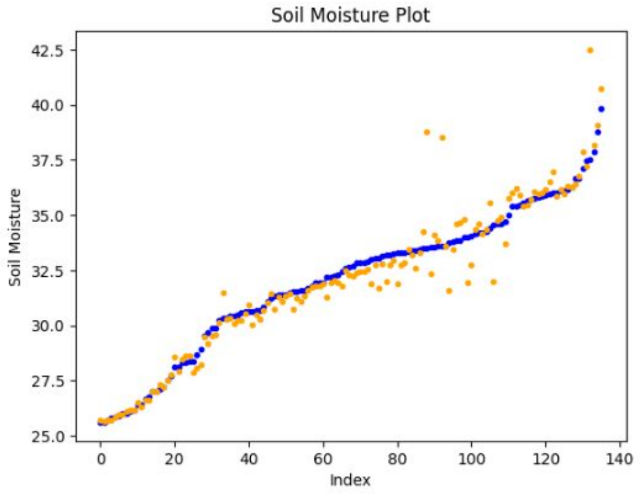




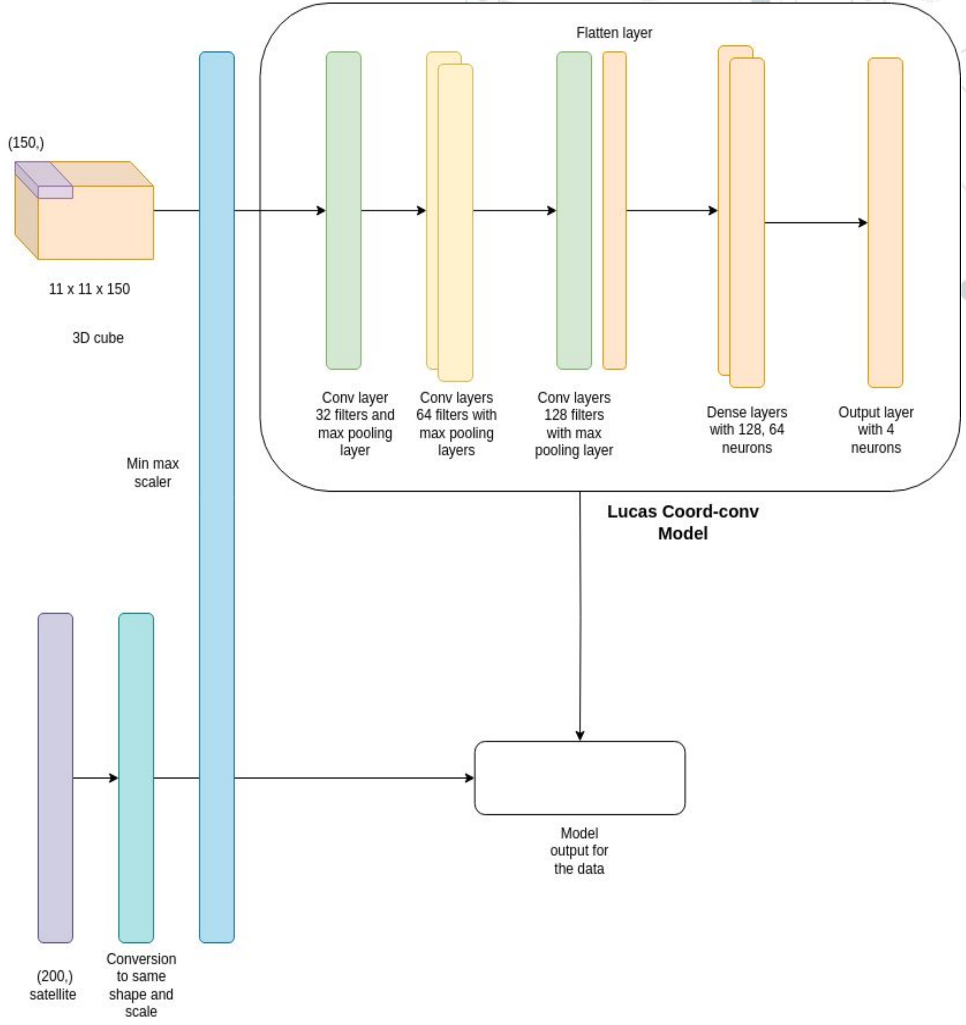
# HIST Gradient

- We got mse of 1.38 and mae of 0.72 while testing for soil temperature. Moreover, while testing for soil moisture we got 0.87 and 0.45 mse and mae respectively.





# Network Architecture


(150,)
11 x 11 x 150
3D cube
Min max scaler
Flatten layer
Conv layer 32 filters and max pooling layer
Conv layers 64 filters with max pooling layers
Conv layers 128 filters with max pooling layer
Dense layers with 128, 64 neurons
Output layer with 4 neurons
Lucas Coord-conv Model
(200,) satellite
Conversion to same shape and scale
Model output for the data

## Satellite Testing

- In our project, we embarked on testing the efficacy of our models using real-time data obtained from the ARSET Satellite. Our primary dataset of interest was the EO-1 Hyperion dataset, renowned for its comprehensive coverage of patches from various geographic regions across the globe spanning the years 2000 to 2017.

- However, despite its extensive coverage, we encountered a significant challenge: the difficulty in accurately identifying patches containing agricultural soil.

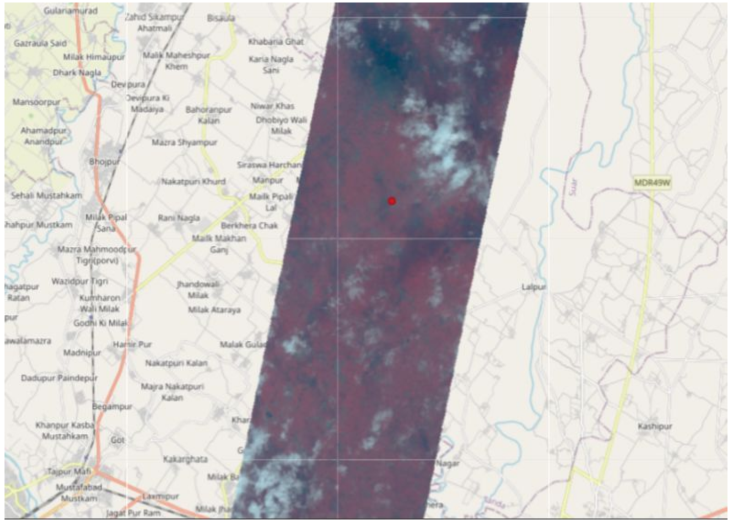
- The models we used were 1D CNN in both cases, they were close to the best results after standardization. Both models and scalers were saved and used for satellite testing.

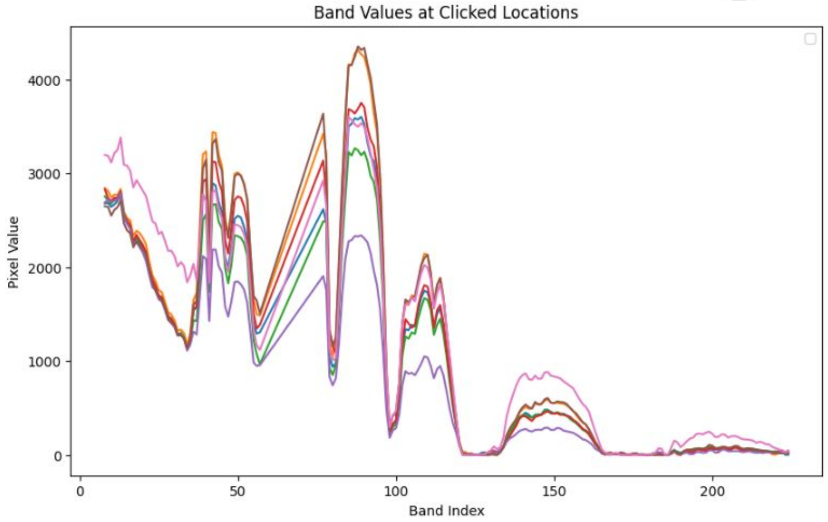
## Demo

- We have designed a code to give us captured patch of the entered area and then allows us to click on the patch and give us band values corresponding to that.

- We also have a webpage connected to api which serves as giving the soil nutrients and soil temperature, soil moisture content of the entered area.

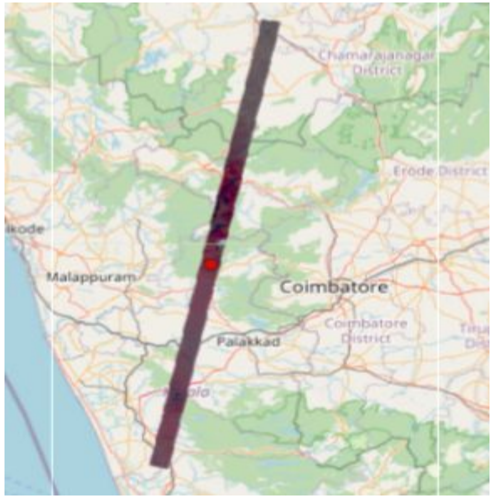
## Demo 1

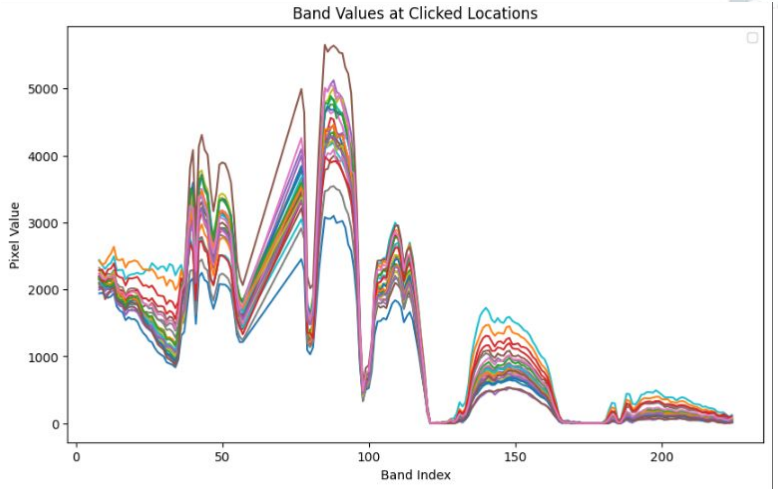




The data from Uttar Pradesh captured by the satellite is mostly of vegetation

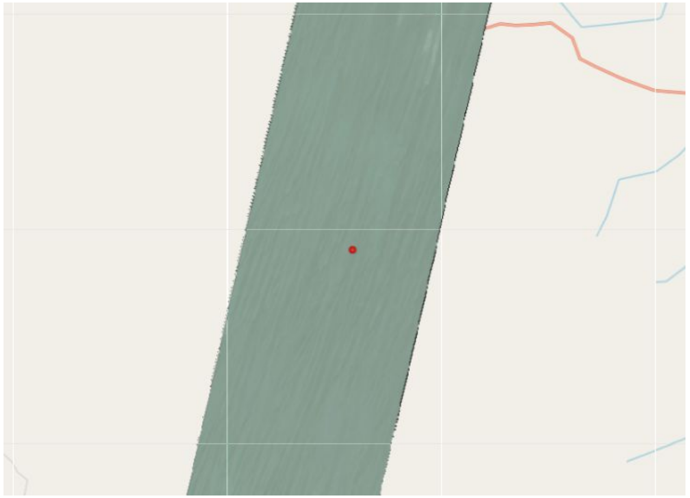
# Demo 1

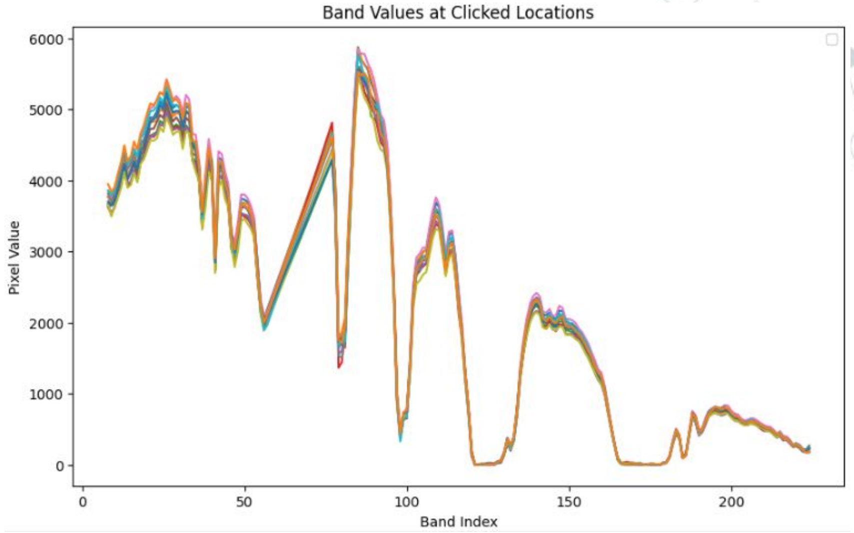




The data from Kerala captured by the satellite is mix of soil data and vegetation

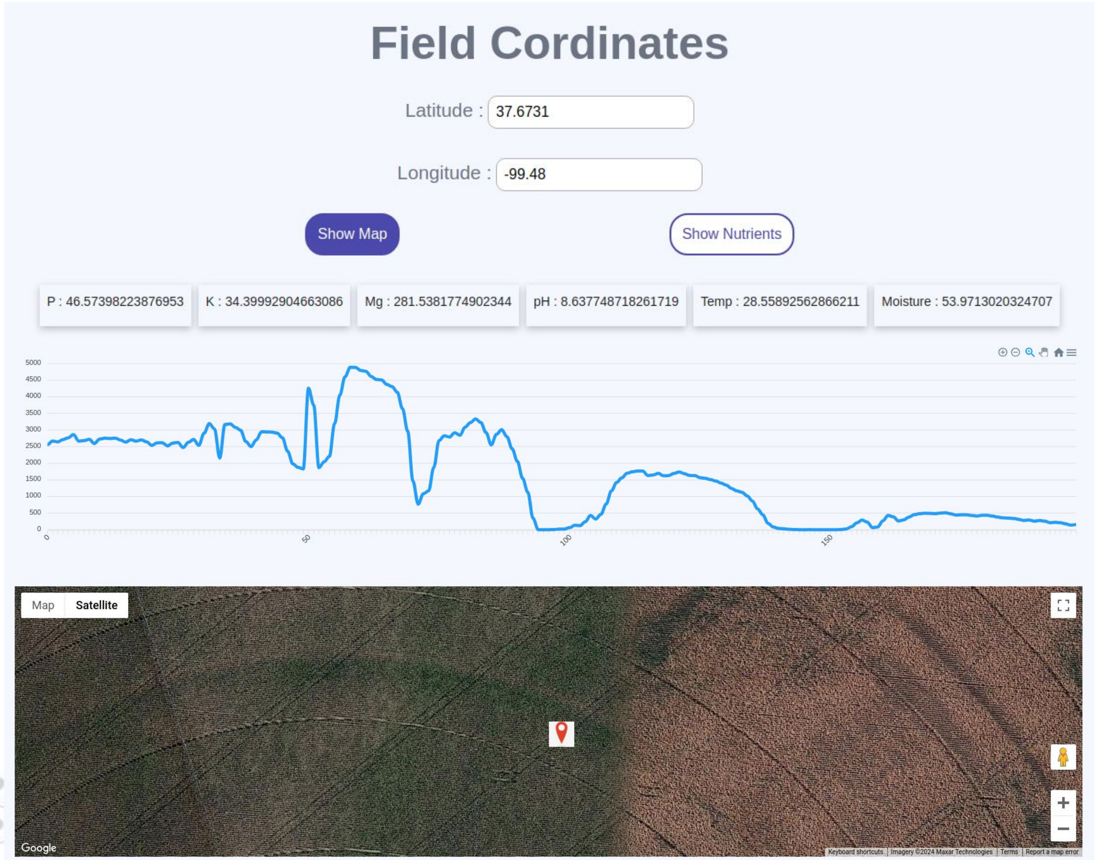
# Demo 1





The data from desert resembling most to be of soil

# Demo 2



Field Cordinates

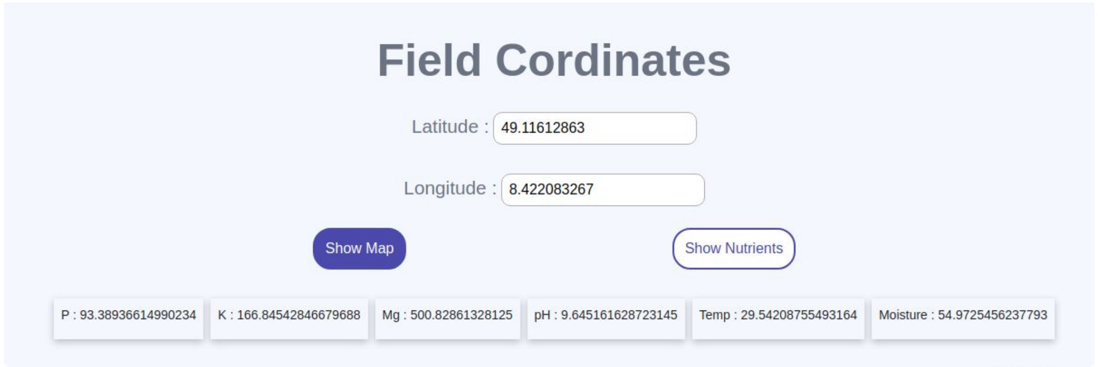
Latitude : 37.6731

Longitude : -99.48

P : 46.57398223876953 | K : 34.39992904663086 | Mg : 281.5381774902344 | pH : 8.637748718261719 | Temp : 28.55892562866211 | Moisture : 53.9713020324707

Field coordinates of corn belt captured in 2007.

## Demo 2

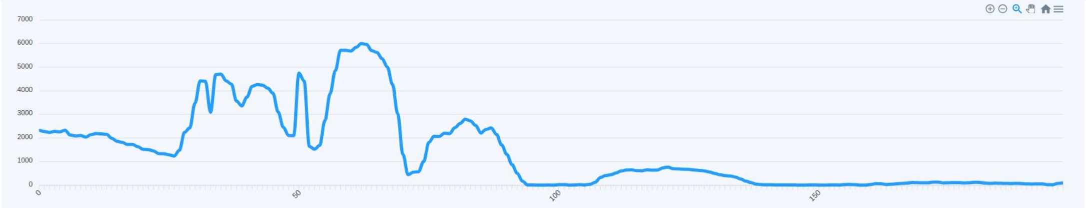


### Field Cordinates

Latitude : 49.11612863

Longitude : 8.422083267

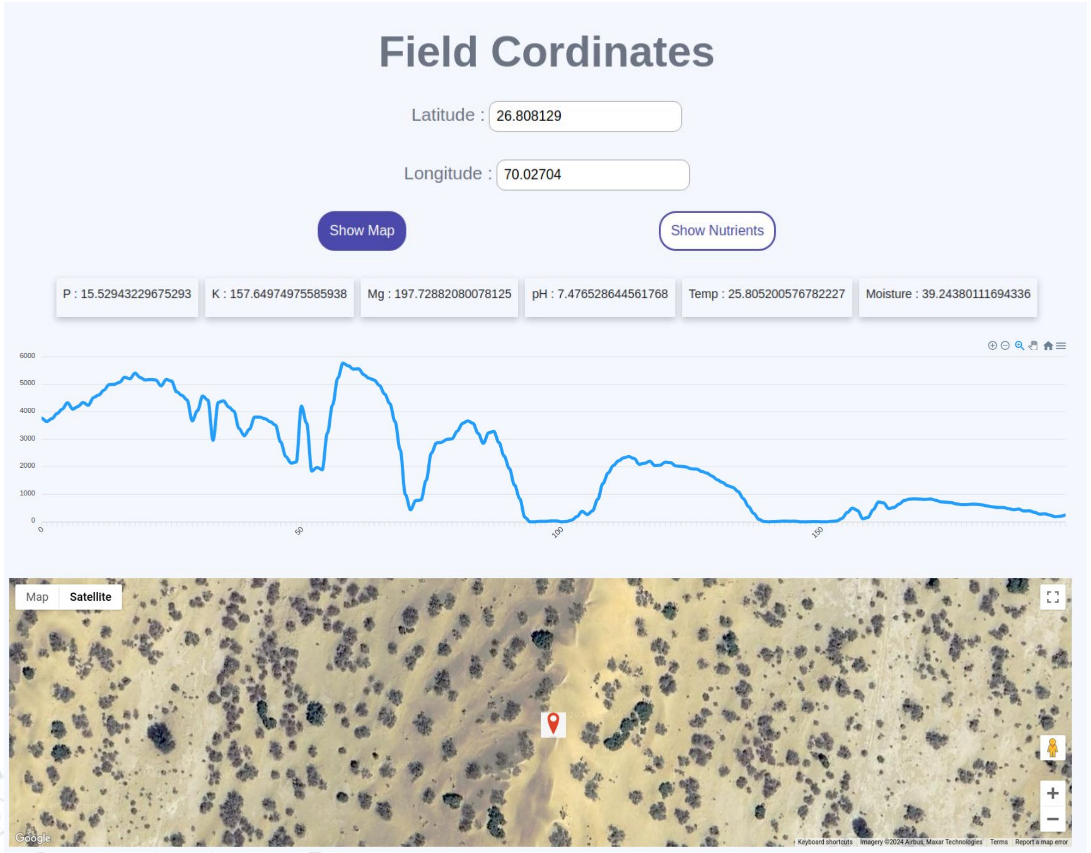
P : 93.38936614990234 | K : 166.84542846679688 | Mg : 500.82861328125 | pH : 9.645161628723145 | Temp : 29.54208755493164 | Moisture : 54.9725456237793



Map
Satellite
Google
Keyboard shortcuts
Imagery ©2024 GeoBasis-DE/BKG, Maxar Technologies
Terms
Report a map error

Field coordinates of lat 49 and long 8.

## Demo 3



# Field Cordinates

Latitude : 26.808129

Longitude : 70.02704

Show Map | Show Nutrients

P : 15.52943229675293 | K : 157.64974975585938 | Mg : 197.72882080078125 | pH : 7.476528644561768 | Temp : 25.805200576782227 | Moisture : 39.24380111694336

Field coordinates of a desert.

## Challenges Faced

- Datasets are gathered from remote locations in a city of Germany and a agricultural field in Poland which makes is quite less to go for checking soil nutrients in entire world on its ground truth.

- Limitations with finding soil patch in the satellite bars us down from checking the quality of soil there.

## Future Work

- Adequate finding technique of soil patch will solve the testing part to much extent and after training on bigger dataset scaling could be modified accordingly.
- Extensive dataset of various locations could be collected and our model seems to be promising when trained on larger areas, and reviewed on global dataset.

# References

- SOIL TEXTURE CLASSIFICATION WITH 1D CONVOLUTIONAL NEURAL, NETWORKS BASED ON HYPERSPECTRAL DATA, F. M. Riese1, S. Keller11 Karlsruhe Institute of Technology (KIT), Institute of Photogrammetry and Remote Sensing,Englerstr. 7, D-76131 Karlsruhe, Germany, (felix.riese, sina.keller)@kit.edu (Link)

- HybridSN: Exploring 3D-2D CNN Feature Hierarchy for Hyperspectral Image Classification Swalpa Kumar Roy, Student Member, IEEE, Gopal Krishna, Shiv Ram Dubey, Member, IEEE, and Bidyut B. Chaudhuri, Life Fellow, IEEE (Link)

- "GeoAI Challenge Estimating Soil Parameters from Hyperspectral Images by ITU" by Zindi, Africa

- Hyperspectral benchmark dataset on soil moisture. Soil moisture data Karlsruhe (Germany), 2017 (kaggle)

- Hyperion Satellite under ARSET (Data)

# Thanks!

**Git :** https://github.com/priyanshu87511/BTP