PROJECT REPORT

On

Forecast Future Pattern of Land cover classes using Time Series Analysis of Digital Image

(CSE V Semester Mini project PCS-504)

2020-2021

Submitted to:

Mr. Samir Rana

(CC-CSE-C-V-Sem)

Guided by:

Dr. Mahesh Manchanda

(Resource Person)

Submitted by:

Shivangi Chauhan

Roll. No.: 1018596

CSE-C-V-Sem

Session: 2020-2021

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Ms. Shivangi Chauhan (Roll No.- 1018596) has developed mini project on "Forecast future patterns of land cover classes using Time Series Analysis of Digital Image" for the CS V Semester Mini Project Lab (PCS-504) in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date: 11/12/2020

(Mr. Samir Rana)	(Dr Mahesh Manchanda)
Project Co-ordinator	**Project Guide**
CC-CSE-C-V-Sem	Resource Person
(CSE Department)	(CSE Department)
GEHU Dehradun	GEHU Dehradun

ACKNOWLEDGMENT

I would like to express our gratitude to **The Almighty Shiva Baba**, the most Beneficent and the most Merciful, for completion of project.

I wish to thank my parents for their continuing support and encouragement. I also wish to thank them for providing us with the opportunity to reach this far in our studies.

I would like to thank particularly our project **Co-ordinator Mr. Samir Rana** and our **Project Guide Dr Mahesh Manchanda** for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last but not the least we greatly indebted to all other persons who directly or indirectly helped us during this work.

SHIVANGI CHAUHAN

Roll No. - 1018596

CSE-C-V-Sem

Session: 2020-2021

GEHU, Dehradun

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 ABOUT PROJECT

Monitoring changes in land usage is an important area of research as land cover is a key variable driving the Earth's energy balance, hydrological and carbon cycle, and the provisioning of natural resources and habitat. (1) Viewing the Earth from space is now crucial to the understanding of the influence of man's activities on his natural resource base over time. In situations of rapid and often unrecorded land use change, observations of the earth from space provide objective information of human utilization of the landscape.

Over the last three decades, satellite based Earth Observation (EO) programs have made tremendous progress in acquiring medium-resolution ($10 - 100$m) images around the globe systematically and with increasing frequency (revisit time). One such is the Copernicus Sentinel-2 mission comprises of a constellation of two polar-orbiting satellites placed in the same sun-synchronous orbit, phased at 180° to each other. It aims at monitoring variability in land surface conditions, and its wide swath width (290 km) and high revisit time (10 days at the equator with one satellite, and 5 days with 2 satellites under cloud-free conditions which results in 2-3 days at mid-latitudes) will support monitoring of Earth's surface changes.

The land use/land cover pattern of a region is an outcome of natural and socio – economic factors and their utilization by man in time and space. Land is becoming a scarce resource due to immense agricultural and demographic pressure. Hence, information on land use / land cover and possibilities for their optimal use is essential for the selection, planning and implementation of land use schemes to meet the increasing demands for basic human needs and welfare. This information also assists in monitoring the dynamics of land use resulting out of changing demands of increasing population.

Land use and land cover change has become a central component in current strategies for managing natural resources and monitoring environmental changes. The advancement in the concept of vegetation mapping has greatly increased research on land use land cover change thus providing an accurate evaluation of the spread and health of the world's forest, grassland, and agricultural resources has become an important priority.

The collection of remotely sensed data facilitates the synoptic analyses of Earth - system function, patterning, and change at local, regional and global scales over time; such data also provide an important link between intensive, localized ecological research and regional, national and international conservation and management of biological diversity. ((Wilkie and Finn)

Therefore, an attempt will be made in this study to construct a machine learning pipeline for predicting the land use and land cover for the region of Republic of Slovenia. Satellite images obtained by ESA's Sentinel-2 to train a model and use it for prediction. (3)

1.2 PROBLEM STATEMENT

Slovenia, a country situated in Central and South-eastern Europe, has witnessed remarkable expansion, growth and developmental activities such as building, road construction, deforestation and many other anthropogenic activities. This has therefore resulted in increased land consumption and a modification and alterations in the status of her land use land cover over time as it changes over time with a view to detecting the land consumption rate and also make attempt to predict same and the possible changes that may occur in this status so that planners can have a basic tool for planning.

1.3 AIM AND OBJECTIVES

The aim of this study is to produce a land use land cover map of Slovenia at different epochs in order to detect the changes that have taken place particularly in the built-up land and subsequently predict likely changes that might take place in the same over a given period.

The following specific objectives will be pursued in order to achieve the aim above.

- To create a land use land cover classification scheme
- To determine the trend, nature, rate, location and magnitude of land use land cover change.
- To forecast the future pattern of land use land cover in the area.
- To generate data on land consumption rate and land absorption coefficient
- since more emphasis is placed on built-up land.
- To evaluate the socio – economic implications of predicted change.

1.4 STUDY AREA

Slovenia lies between latitudes 45° and 47° N, and longitudes 13° and 17° E. On the Pannonian plain to the East and Northeast, toward the Croatian and Hungarian borders, the landscape is essentially flat. However, most of Slovenia is hilly or mountainous, with around 90% of its land surface 200 m (656 ft) or more above sea level.

Over half of the country (11,823 km^2 or 4,565 sq mi) is covered by forests. This makes Slovenia the third most forested country in Europe, after Finland and Sweden. The areas are covered mostly by beech, fir-beech and beech-oak forests and have a relatively high production capacity. Remnants of primeval forests are still to be found, the largest in the Kočevje area. Grassland covers 5,593 km^2 (2,159 sq mi) and fields and gardens (954 km^2 or 368 sq mi). There are 363 km^2 (140 sq mi) of orchards and 216 km^2 (83 sq mi) of vineyards. (7)

Slovenia is in a rather active seismic zone because of its position on the small Adriatic Plate, which is squeezed between the Eurasian Plate to the north and the African Plate to the south and rotates counter-clockwise. Thus the country is at the junction of three important geotectonic units: the Alps to the north, the Dinaric Alps to the south and the Pannonian Basin to the east Scientists have been able to identify 60 destructive earthquakes in the past. Additionally, a network of seismic stations is active throughout the country.

CHAPTER 2

PROJECT

2.1 INTRODUCTION

The procedure adopted in this project work predicts the future classes of land use dynamics of a digital image.

2.2 DATA SOURCE

For the study, Landsat satellite images of Republic of Slovenia are public. The digital image was acquired from ESA's Sentinel-2 in three Epochs; 1972, 1986 and 2001.

We will use satellite images obtained by ESA's Sentinel-2 to train a model and use it for prediction. This study will lead through the whole process of creating the pipeline, with details provided at each step complement number bitwise. (6)

CODE	LAND USE/LAND COVER CATEGORIES
1	Farmland
2	Wasteland
3	Built-up land
4	Forestland
5	Water bodies

Table 2.1 Land use land cover classification scheme

The definition of waste land as used in this research work denotes land without scrub, sandy areas, dry grasses, rocky areas and other human induced barren lands.

2.3 LIMITATIONS

- Labels are determined at specific time, but land use can change (what was once a field, may now be a house)
- Labels are overly generalized (a city is an artificial surface, but it also contains parks, forests etc.)
- Some classes can have an overlap or similar definitions (part of a continuum, and not discrete distributions)
- Human error (mistakes made when producing the reference map)

2.4 METHODS OF ANALYSIS

Workflow that processes the data:

- Remove too cloudy scenes

- Check the ratio of the valid data for each patch and for each time frame
- Keep only time frames with > 80 % valid coverage (no clouds)
- Concatenate BAND, NDVI, NDWI, NDBI info into a single feature called Perform temporal interpolation (filling gaps and resampling to the same dates)
- Create a task for linear interpolation in the temporal dimension
- Provide the cloud mask to tell the interpolating function which values to update
- Perform erosion
- This removes artefacts with a width of 1 px, and also removes the edges between polygons of different classes
- Random spatial sampling of the EOPatches
- Randomly take a subset of pixels from a patch to use in the machine learning training
- Split patches for training/validation
- Split the patches into a training and validation set

Also, a subset of the country-wide reference for land-use-land-cover is provided. It is available in the form of a geopackage, which contains polygons and their corresponding labels. The labels represent the following 10 classes:

- lulcid = 0, name = no data
- lulcid = 1, name = cultivated land
- lulcid = 2, name = forest
- lulcid = 3, name = grassland
- lulcid = 4, name = shrubland
- lulcid = 5, name = water
- lulcid = 6, name = wetlands
- lulcid = 7, name = tundra
- lulcid = 8, name = artificial surface
- lulcid = 9, name = bareland
- lulcid = 10, name = snow and ice

The test sample is hand picked, has small set of patches, otherwise with a larged overall set, the training and testing patches should be randomly chosen.

The sampled features and labels are loaded and reshaped into $n \times m$, where n represents the number of training pixels, and $m = f \times t$ the number of all features (in this example 216), with f the size of bands and band combinations (in this example 9) and t the length of the resampled time-series (in this example 25)

LightGBM is used as a ML model. It is a fast, distributed, high performance gradient boosting framework based on decision tree algorithms, used for many machine learning task.

2.5 LAND USE LAND COVER DISTRIBUTION

The static land use land cover distribution for each study year as derived from the maps are presented in the table below

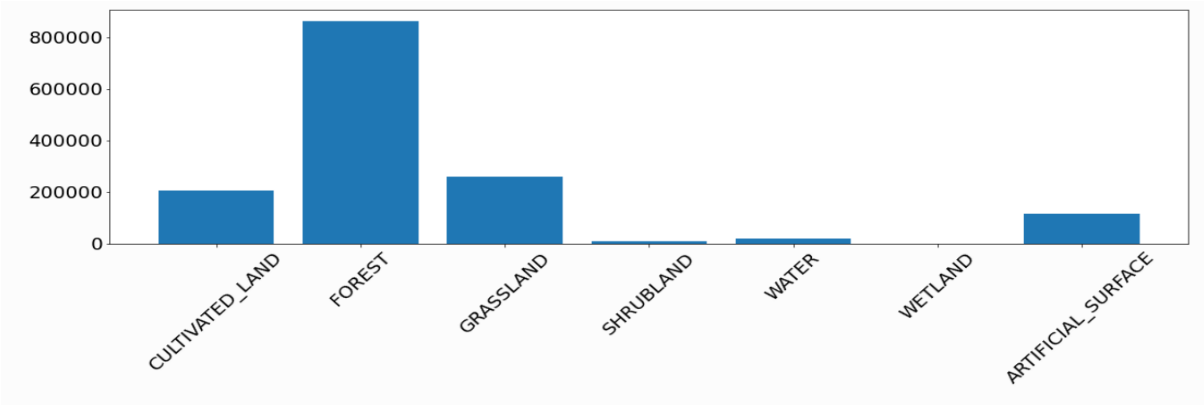


Figure 2.1 Land Cover Distribution

2.6 LAND USE AND LAND COVER DISTRIBUTION

EOPatches were created and filled with Sentinel-2 data using Sentinel Hub services. Following data was added to each EOPatch:

L1C custom list of bands [B02, B03, B04, B08, B11, B12], which corresponds to [B, G, R, NIR, SWIR1, SWIR2] wavelengths.

Additionally, add:

- Calculated NDVI, NDWI, and NDBI information
- A mask of validity, based on acquired data from Sentinel and cloud coverage. Valid pixel is if:
- IS_DATA == True
- CLOUD_MASK == 0 (1 indicates cloudy pixels and 255 indicates NO DATA)

An EOPatch is created and manipulated using EOTasks, which are chained in an EOWorkflow. In this study the final workflow is executed on all patches, which are saved to the specified directory.

Due to the filtering, not all patches have the same amount of timestamps. A date was selected and the closest timestamp for each eopatch was drawn.

2.6.1 SPATIAL MEAN OF NDVI

Plot the mean of NDVI over all pixels in a selected patch throughout the year. Filter ` out clouds in the mean calculation.

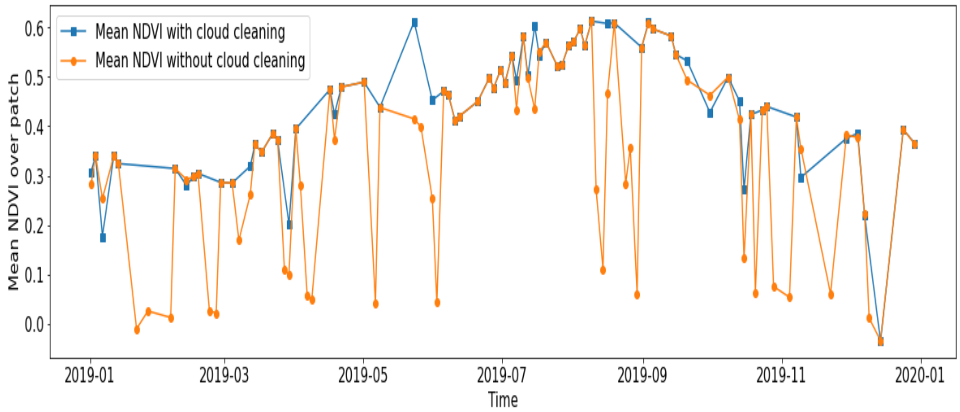


Fig 2.2 NDVI over all pixels of a selected patch

2.7 PREARE TRAINING DATA

We will create a new workflow that processes the data:

- Remove too cloudy scenes
- Check the ratio of the valid data for each patch and for each time frame
- Keep only time frames with > 80 % valid coverage (no clouds)
- Concatenate BAND, NDVI, NDWI, NDBI info into a single feature called FEATURES
- Perform temporal interpolation (filling gaps and resampling to the same dates)
- Create a task for linear interpolation in the temporal dimension
- Provide the cloud mask to tell the interpolating function which values to update
- Perform erosion
- This removes artefacts with a width of 1 px, and also removes the edges between polygons of different classes
- Random spatial sampling of the EOPatches
- Randomly take a subset of pixels from a patch to use in the machine learning training
- Split patches for training/validation
- Split the patches into a training and validation set

2.7.1 WORKFLOW

All the tasks that were defined so far create and fill the EOPatches. The tasks need to be put in some order and executed one by one..The following workflow is created and executed:

- Create EOPatches with band and cloud data
- Calculate and add NDVI, NDWI, NORM
- Add mask of valid pixels
- Add scalar feature representing the count of valid pixels
- Save eopatches

2.8 MODEL CONSTRUCTION AND TRAINING

The patches are split into a train and test subset, where we split the patches for training and testing.

The test sample is hand picked because of the small set of patches, otherwise with a larged overall set, the training and testing patches should be randomly chosen.

The sampled features and labels are loaded and reshaped into n×mn×m, where nn represents the number of training pixels, and m=f×tm=f×t the number of all features (in this example 216), with ff the size of bands and band combinations (in this example 9) and tt the length of the resampled time-series.

LightGBM is used as a ML model. It is a fast, distributed, high performance gradient boosting framework based on decision tree algorithms, used for many machine learning tasks. (5)

Classification accuracy 93.7%
Classification F1-score 93.6%

Class	F1	Recall	Precision
* CULTIVATED_LAND	88.1	86.1	90.2
* FOREST	98.6	98.8	98.4
* GRASSLAND	87.8	89.0	86.6
* SHRUBLAND	35.3	32.4	38.8
* WATER	95.1	98.0	92.3
* ARTIFICIAL_SURFACE	95.6	96.0	95.2

Table 2.2 Classification accuracy and F1 scores

2.8.1 ROC CURVES AND AUC

Calculate precision and recall rates, draw ROC curves and calculate AUC.

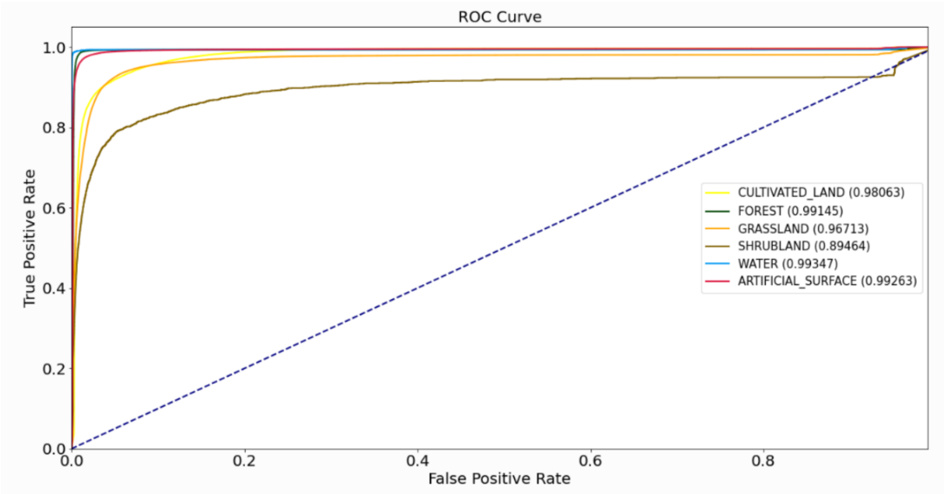


Figure 2.3 ROC Curve plot

2.8.2 CONFUSION MATRIX

For most of the classes the model seems to perform well. Otherwise the training sample is probably too small to make a fair assesment. Additional problems arise due to the unbalanced training set. The image below shows the frequency of the classes used for model training, and we see that the problematic cases are all the under-represented classes: shrubland, water, wetland, and bareland.Improving the reference map would also affect the end result, as, for example some classes are mixed up to some level.

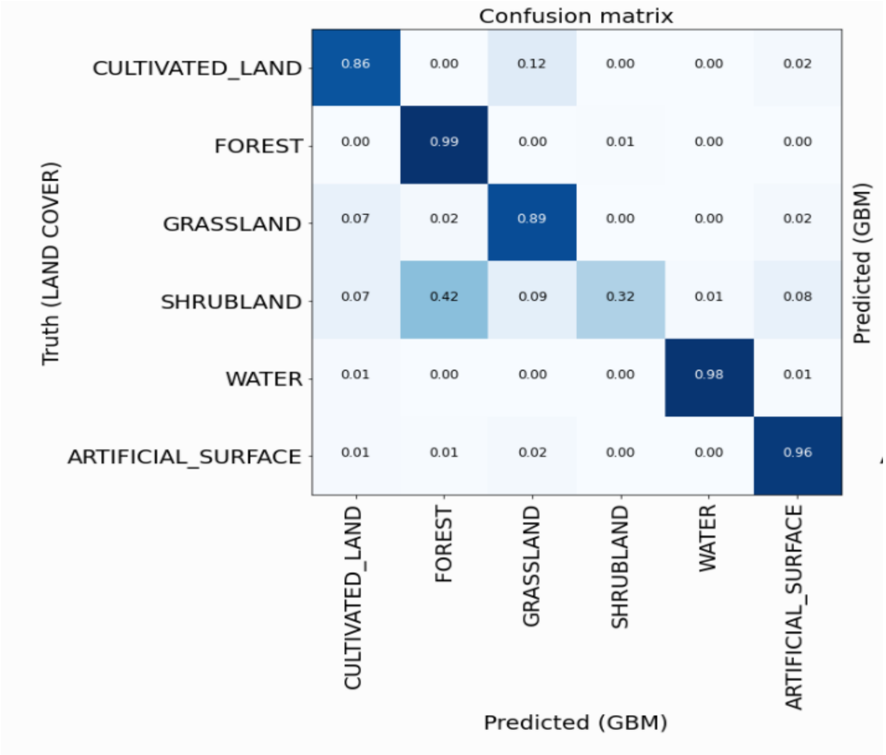


Figure 2.4 Confusion Matrix

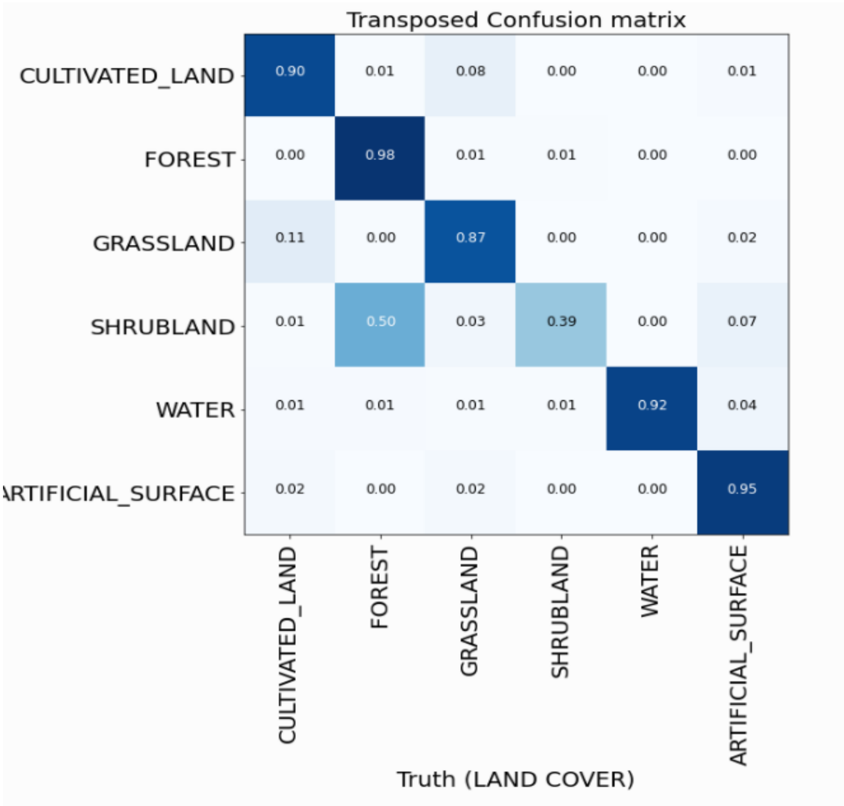


Figure 2.5 Transposed Confusion Matrix

2.9 PREDICTIONS

Random subsets of patches are chosen, where prediction and ground truth are compared.

For visual aid the mask of differences and the true color image are also provided.

In majority of the cases, differences seem to lie on the border of different structures.

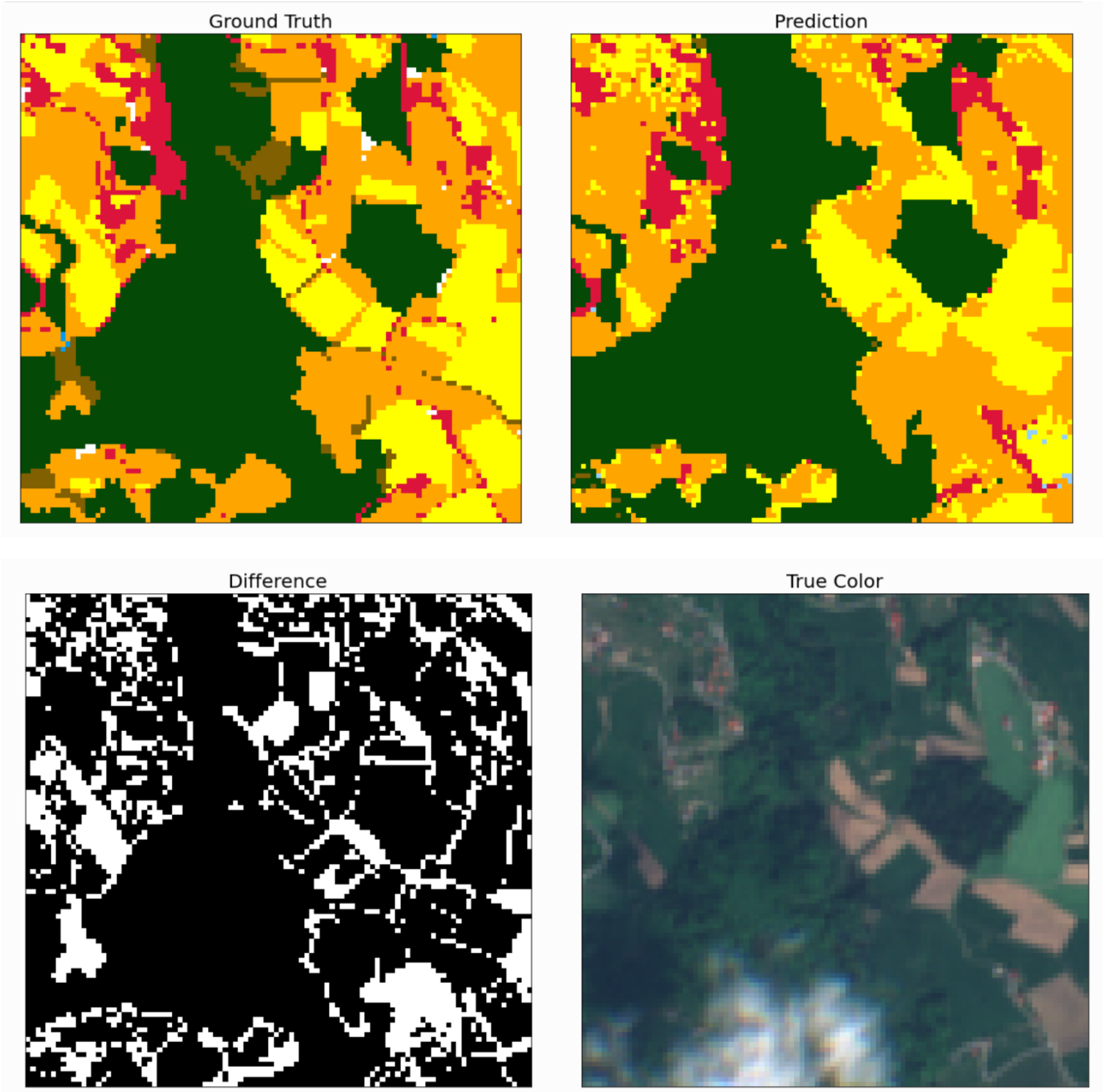


Figure 2.6 Prediction vs Ground Truth

CHAPTER 3

SNAPSHOT OF PROJECT

3.1 GET COUNTRY BOUNDARY

```
[2]: # Folder where data for running the notebook is stored
     DATA_FOLDER = os.path.join('..', '..', 'example_data')

     # Load geojson file
     country = gpd.read_file(os.path.join(DATA_FOLDER, 'svn_border.geojson'))
     country = country.buffer(500)

     # Get the country's shape in polygon format
     country_shape = country.geometry.values[-1]

     # Plot country
     country.plot()
     plt.axis('off');

     # Print size
     print('Dimension of the area is {0:.0f} x {1:.0f} m2'.format(country_shape.bounds[2] - country_shape.bounds[0],
                                                                 country_shape.bounds[3] - country_shape.bounds[1]))
```

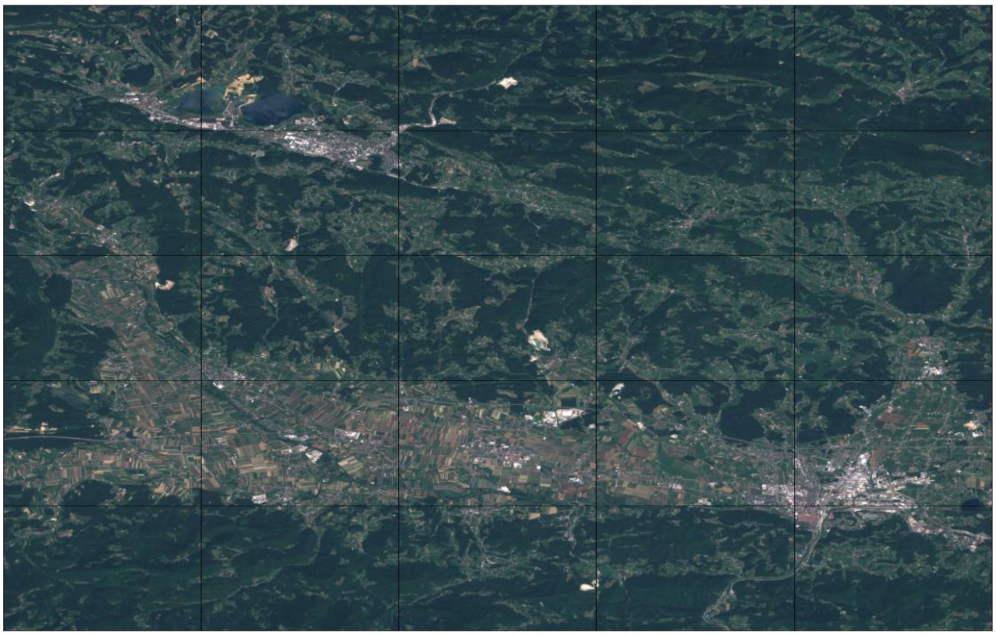
```
Dimension of the area is 243184 x 161584 m2
```

3.2 SLICING INTO 5x5 TILES

```
[4]: # figure
     fig, ax = plt.subplots(figsize=(30, 30))
     gdf.plot(ax=ax,facecolor='w',edgecolor='r',alpha=0.5)
     country.plot(ax=ax, facecolor='w',edgecolor='b',alpha=0.5)
     ax.set_title('Selected 5x5  tiles from Slovenia', fontsize=25);
     for bbox, info in zip(bbox_list, info_list):
         geo = bbox.geometry
         ax.text(geo.centroid.x, geo.centroid.y, info['index'], ha='

     gdf[gdf.index.isin(patchIDs)].plot(ax=ax,facecolor='g',edgecolo

     plt.axis('off');
```

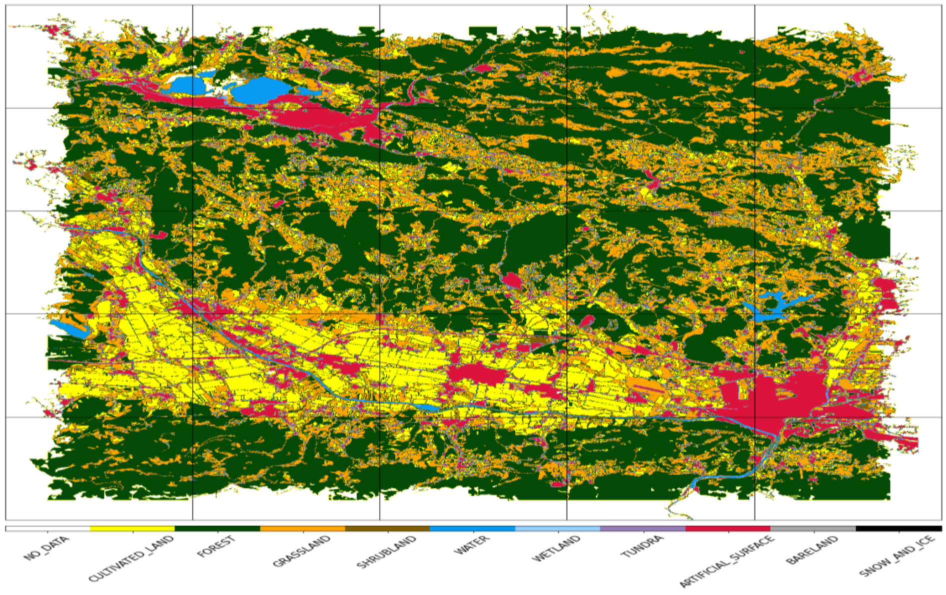
3.3VISUALISE TIMESTAMP FOR EACH PATCH

```
# Draw the RGB image
path_out = './eopatches'
fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 20))

date = datetime.datetime(2019,7,1)

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    dates = np.array(eopatch.timestamp)
    closest_date_id = np.argsort(abs(date-dates))[0]
    ax.imshow(np.clip(eopatch.data['BANDS'][closest_date_id][..., [2, 1, 0]] * 3.5, 0, 1))
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect("auto")
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)
```



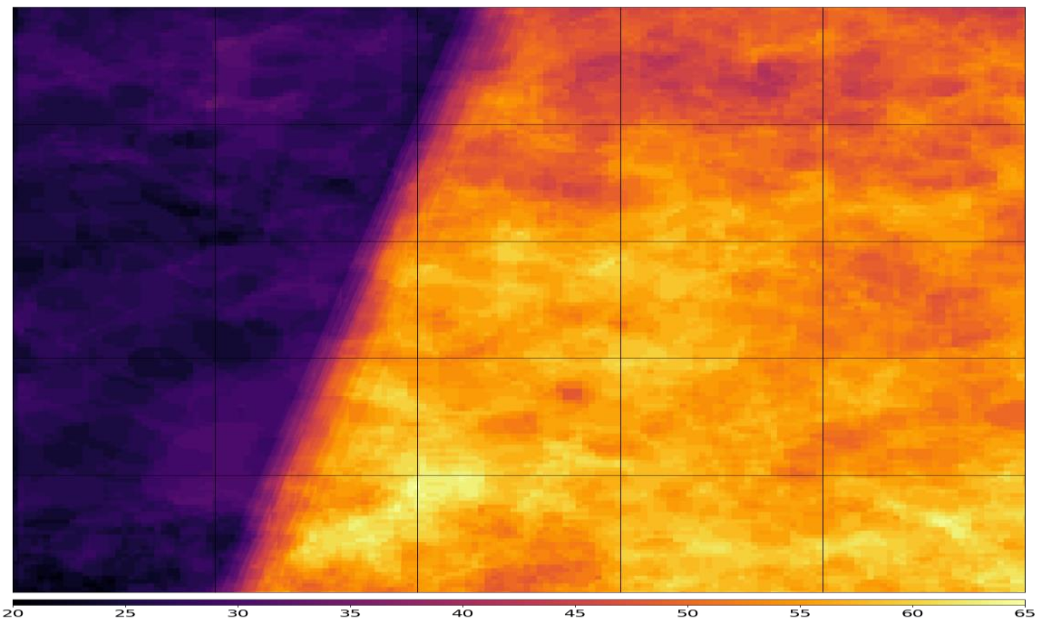
3.4VISUALIZE REFERENCE MAP

```
path_out = './eopatches'

fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 25))

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    im = ax.imshow(eopatch.mask_timeless['LULC'].squeeze(), cmap=lulc_cmap, norm=lulc_norm)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect('auto')
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=axs.ravel().tolist(), orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
cb.set_ticks([entry.id for entry in LULC])
cb.ax.set_xticklabels([entry.name for entry in LULC], rotation=45, fontsize=15)
plt.show()
```



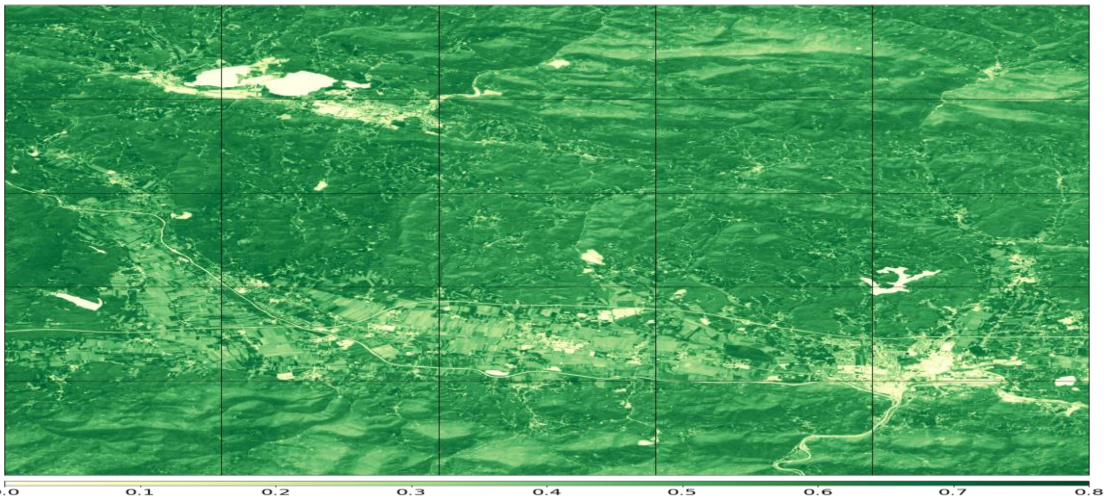
3.5PLOT MAP OF VALID PIXELS COUNT

```
path_out = './eopatches'

vmin, vmax = None, None
for i in range(len(patchIDs)):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    data = eopatch.mask_timeless['VALID_COUNT'].squeeze()
    vmin = np.min(data) if vmin is None else (np.min(data) if np.min(data) < vmin else vmin)
    vmax = np.max(data) if vmax is None else (np.max(data) if np.max(data) > vmax else vmax)

fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 25))

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    im = ax.imshow(eopatch.mask_timeless['VALID_COUNT'].squeeze(), vmin=vmin, vmax=vmax, cmap=plt.cm.inferno)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect("auto")
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=axs.ravel().tolist(), orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
plt.show()
```



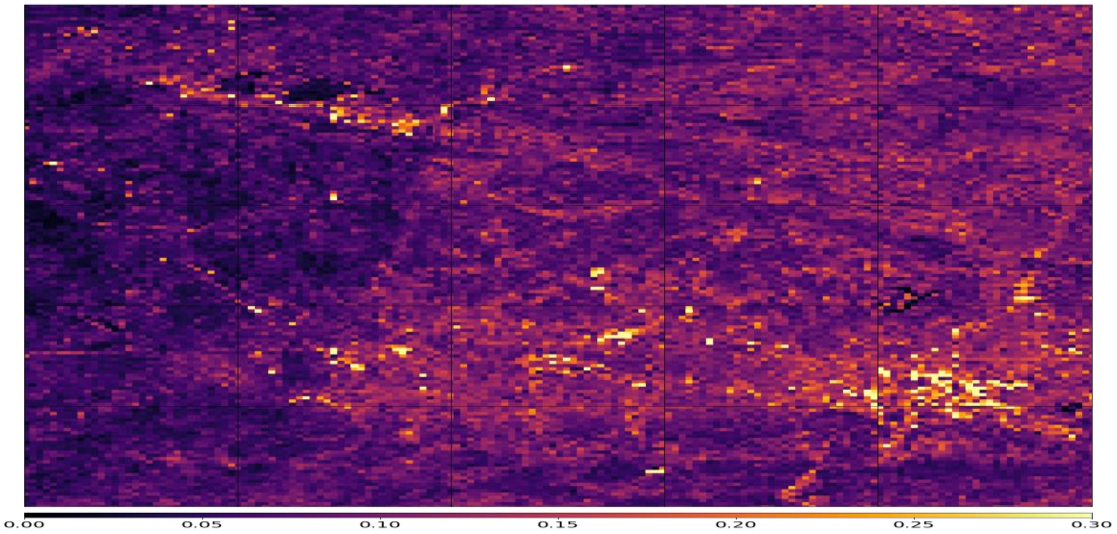
3.6TEMPERAL MEAN OF NDVI

[15]:
```
path_out = './eopatches'

fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 25))

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    ndvi = eopatch.data['NDVI']
    mask = eopatch.mask['IS_VALID']
    ndvi[~mask] = np.nan
    ndvi_mean = np.nanmean(ndvi, axis=0).squeeze()
    im = ax.imshow(ndvi_mean, vmin=0, vmax=0.8, cmap=plt.get_cmap('YlGn'))
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect("auto")
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=axs.ravel().tolist(), orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
plt.show()
```



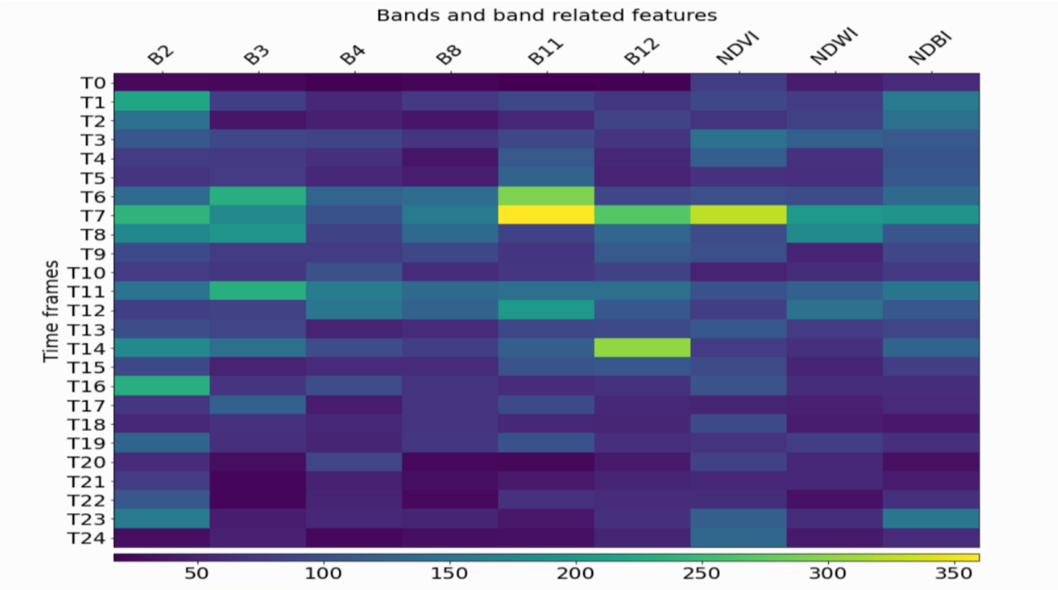
3.7CLOUD PROBABILITY

[16]:
```
path_out = './eopatches'

fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 25))

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    clp = eopatch.data['CLP'].astype(float)/255
    mask = eopatch.mask['IS_VALID']
    clp[~mask] = np.nan
    clp_mean = np.nanmean(clp, axis=0).squeeze()
    im = ax.imshow(clp_mean, vmin=0.0, vmax=0.3, cmap=plt.cm.inferno)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect("auto")
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=axs.ravel().tolist(), orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
plt.show()
```



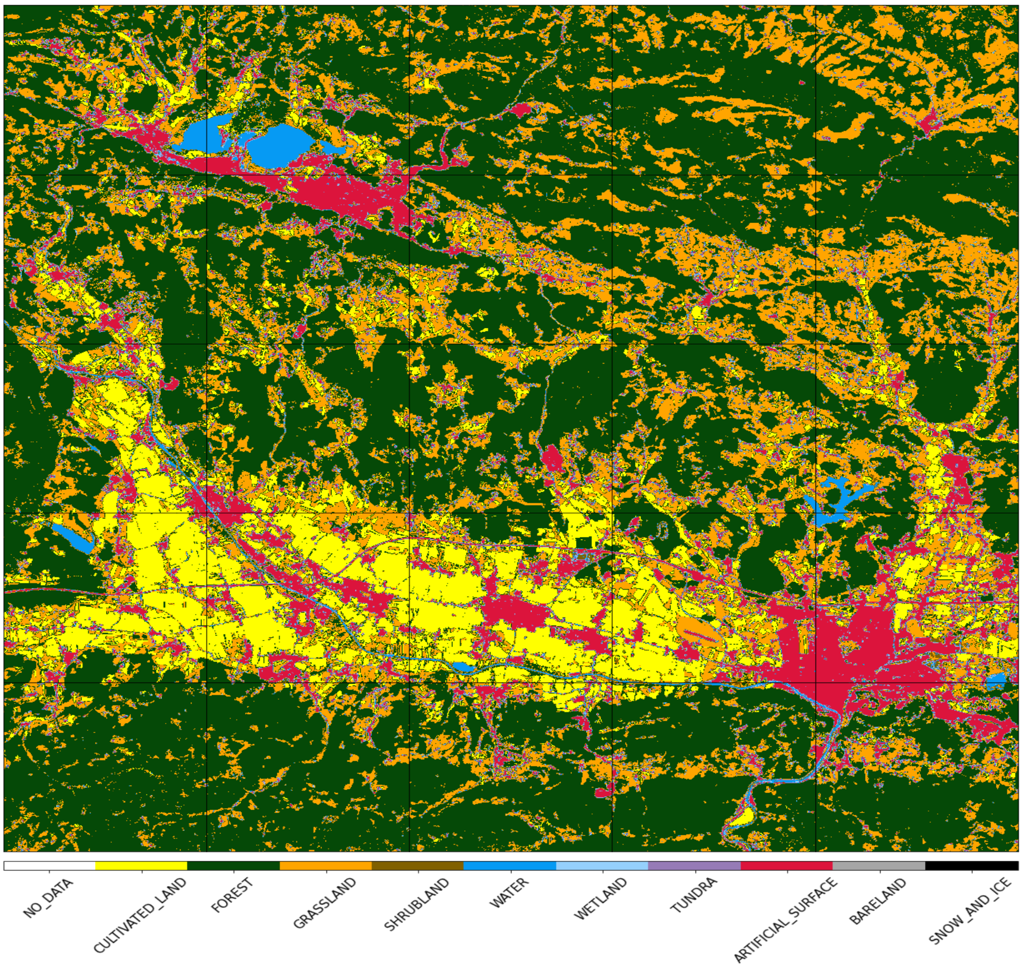
3.8 FEATURES USING LIGHTGBM MODEL

```
# names of features
fnames = ['B2','B3','B4','B8','B11','B12','NDVI','NDWI','NDBI']

# get feature importances and reshape them to dates and features
feature_importances = model.feature_importances_.reshape((t, f))

fig = plt.figure(figsize=(15, 15))
ax = plt.gca()

# plot the importances
im = ax.imshow(feature_importances, aspect=0.25)
plt.xticks(range(len(fnames)), fnames, rotation=45, fontsize=20)
plt.yticks(range(t), [f'T{i}' for i in range(t)], fontsize=20)
plt.xlabel('Bands and band related features', fontsize=20)
plt.ylabel('Time frames', fontsize=20)
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=[ax], orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
```



3.9 VISUALISE PREDICTION

```
path_out_sampled = './eopatches_sampled'

fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(20, 25))

for i in tqdm(range(len(patchIDs))):
    eopatch = EOPatch.load(f'{path_out_sampled}/eopatch_{i}', lazy_loading=True)
    ax = axs[i//5][i%5]
    im = ax.imshow(eopatch.mask_timeless['LBL_GBM'].squeeze(), cmap=lulc_cmap, norm=lulc_norm)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_aspect("auto")
    del eopatch

fig.subplots_adjust(wspace=0, hspace=0)

cb = fig.colorbar(im, ax=axs.ravel().tolist(), orientation='horizontal', pad=0.01, aspect=100)
cb.ax.tick_params(labelsize=20)
cb.set_ticks([entry.id for entry in LULC])
cb.ax.set_xticklabels([entry.name for entry in LULC], rotation=45, fontsize=15)
plt.show()
```



CHAPTER 4

CONCLUSION

4.1 CONCLUSION

This project demonstrates the ability of time series analysis in capturing spatial-temporal data. Attempt was made to capture as accurate as possible five land use land cover classes as they change through time. In achieving this, Land Consumption Rate and Land Absorption Coefficient were introduced into the study work. The trained model was used to predict the labels on the test sample and the obtained results were then validated against the ground-truth. Overall accuracy of 84.4% and a weighted F1 score of 85.4% were achieved.

4.2 FUTURE SCOPE

These results represent preliminary work on a prototype architecture, which was not optimised for the task at hand. Despite this, results are in line with some of the reported works in the field. Future scope lies in the optimisation of the architecture (e.g. number of features, depth of the network, number of convolutions) and of the hyper-parameters (e.g. learning rate, number of epochs, class weighting) is required to fully assess the potential of TFCNs (4)

References

(n.d.). http://www.naturalearthdata.com/downloads/10m-cultural-vectors/.

(Wilkie and Finn, 1. (n.d.).

https://en.wikipedia.org/wiki/Slovenia#Geography. (n.d.).

https://eo-learn.readthedocs.io/en/latest/examples/land-cover-map/SI_LULC_pipeline.html. (n.d.).

https://medium.com/sentinel-hub/land-cover-classification-with-eo-learn-part-2-bd9aa86f8500. (n.d.).

J. Ross Quinlan. 1986. Induction of decision trees. Machine learning 1, 1. (. (n.d.).

land, M. H. (n.d.).

(n.d.). *Monitoring changes in land usage is an important area of research.*