

**INDIAN INSTITUTE OF REMOTE SENSING**  
**Indian Space Research Organization**  
**Department of space, Government of India**



**DOCUMENTATION**

***"Implementation of ANN/CNN and concepts of Deep Learning in Image-Classification in Remote Sensing"***

**June-July, 2018**

**Under the guidance of**

*Dr. Anil Kumar  
Head PRSD, Scientist/Engineer 'SG'  
PRSD/IIRS/ISRO*

**Submitted By:**

Amolak Bahri (Roll no: 15103020)  
Kanu Priya (Roll no: 15103045)  
Shubham Singh (Roll no: 15103038)  
Vishal Gautam (Roll no: 15103047)  
Rahul Kumar (Roll no: 15103049)  
Mohit meena (Roll no: 15103069)

## **INTRODUCTION:**

The image classification is done on the given BIL-image dataset “apex12bands” of 8 bands, 12 bands and 17 bands using the concepts of ANN/CNN and deep learning. The implementation is done on the Python 3 using the editor Anaconda. The image is classified into 7 clusters namely grasslands, building1, building2, bright forest, dull forest, river and road. Encouraging results of 94.21% accuracy is achieved as a result of fuzzy classification.

**REQUIREMENTS:** Knowledge of ANN/CNN, Basics of Deep learning, python 3

**MODULES:** TensorFlow, Keras, GDAL, numpy, learning, matplotlib

**SOFTWARE USED:** Anaconda, Jupyter notebook.

## **EXPLANATION:**

The image classification is done on the dataset provided and has been tested against different bands i.e. 8, 12 and 17 bands.

The dataset was imaged in the vicinity of Baden, Switzerland on a clear day, with the sensor mounted on a Dornier DO-228 aircraft.



Image Interpretation includes:

1. Training the classifiers
2. Testing the dataset

The classes were identified from the interpretation from the image provided. Regions of Interest were collected, verified with the data providers and used for classification and accuracy analysis.

Given below are the Identified Classes from the input given using soft classification methods stored in the array named **path**

```
path = ["grass_17", "build1_17", "build2_17", "Bright17", "Dull17", "river_17", "road_17"]
```

### Sequence of Implementation

- Import the libraries and their modules: numpy, matplotlib, gdal, keras
- Read the BIL file for input
- Define the training data and read the data from BIL file
- Define xtest,ytest,path and initial values for all the classes to be classified on the basis of model being run.
- Calculate all the clicks and store in clicks{ } array.
- Swap little-endian and Big-Endian.
- Calculate the length of all values
- Define the training data
- Print xtest, xtrain, ytest, ytrain
- Define the model: `model = Sequential()`
- Add the convolution, MaxPooling and dense (fully-connected) layers

```
# model.add(Dense(num_pixels, input_dim=num_pixels,
kernel_initializer='normal', activation='relu'))
model.add(Conv1D(2 ** 2, 2, activation="relu", padding='same',
input_shape=[12, 1]))
model.add(Conv1D(2 ** 3, 2, activation="relu", padding='same'))
model.add(Conv1D(2 ** 4, 2, activation="relu", padding='same'))
model.add(MaxPooling1D(2))
model.add(Conv1D(2 ** 5, 2, activation="relu", padding='same'))
model.add(Conv1D(2 ** 6, 2, activation="relu", padding='same'))
# model.add(MaxPooling1D(2))
model.add(Conv1D(2 ** 7, 2, activation="relu", padding='same'))
model.add(Conv1D(2 ** 8, 2, activation="relu", padding='same'))
```

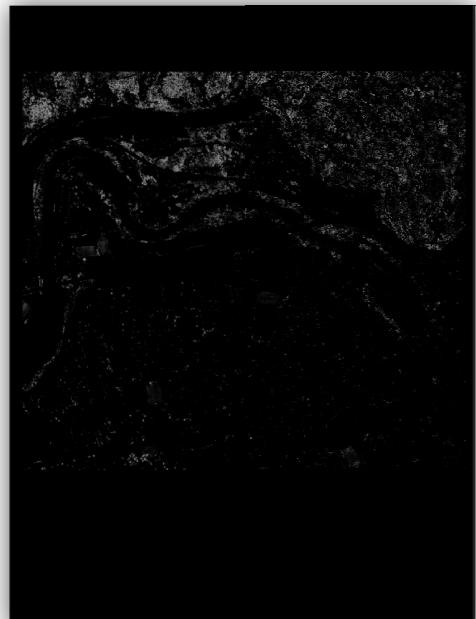
- Run the model as many times as epochs is defined in batch size.
- Predict the model
- Save images in specified folders : images are saved in .tiff format

## OUTPUT:

For Soft Classification:



Road class



Dull-Forest class



Building\_1 class



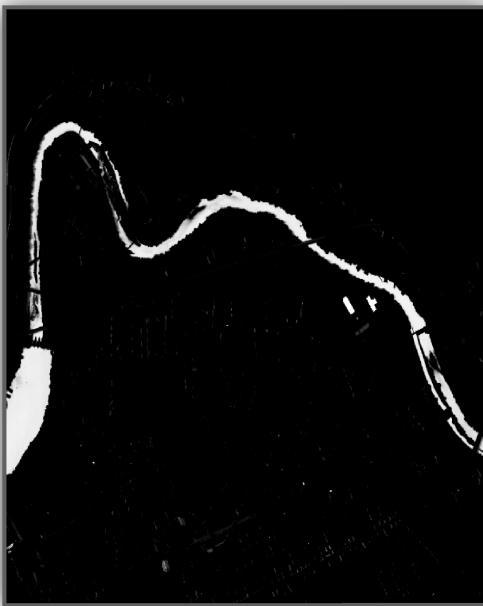
Building\_2 class



Bright forest class

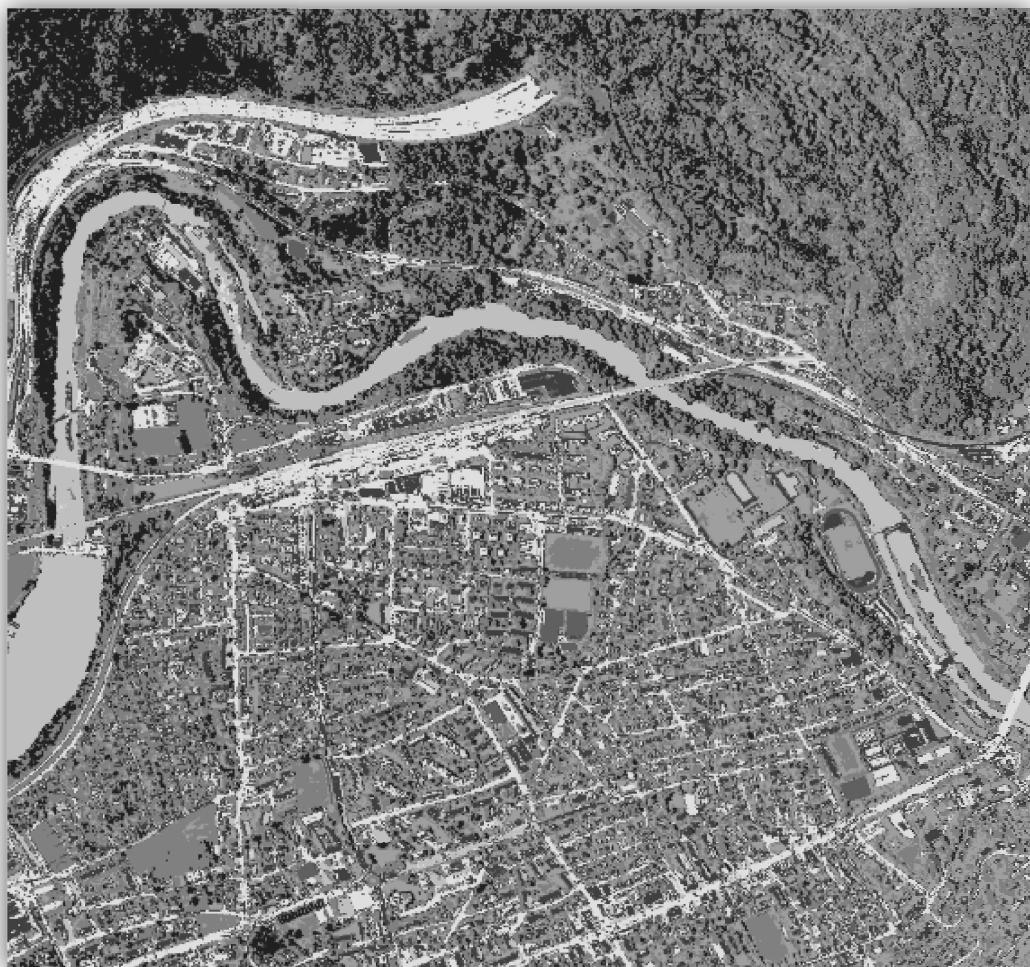


Grassland class



River class

## **For Hard Classification:**



Hard classification of the image

## **CONCLUSION:**

On running this project the given image dataset “apex12 bands” of 8 bands, 12 bands and 17 bands is classified based on the files defined in the path array i.e. Roads, dull forest, building\_1, building\_2, bright forest, grasslands and river.

The classification is done on the basis of soft and hard classification and resulting images are saved in the defined folders in .tiff format.