INDIAN INSTITUTE OF REMOTE SENSING

Indian Space Research Organization

Departmen t of space, Government of India

DOCUMENTATION

“Implementation of RNN and 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:
Lakshya Dhingra
Pawandeep Singh
Pratyush Jha

Pranav Negi
Gurdeep Singh
Vinay Sahanan

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 RNN 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/RNN, Basics of Deep leaning, 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.

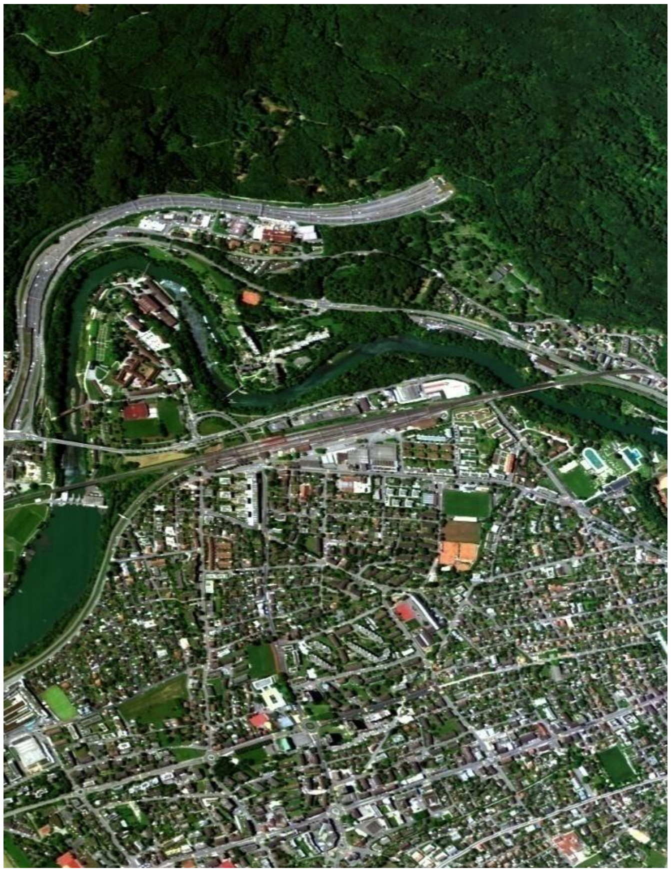


Image Interpretation includes:

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 LSTM, MaxPooling and dense (fully-connected) layers

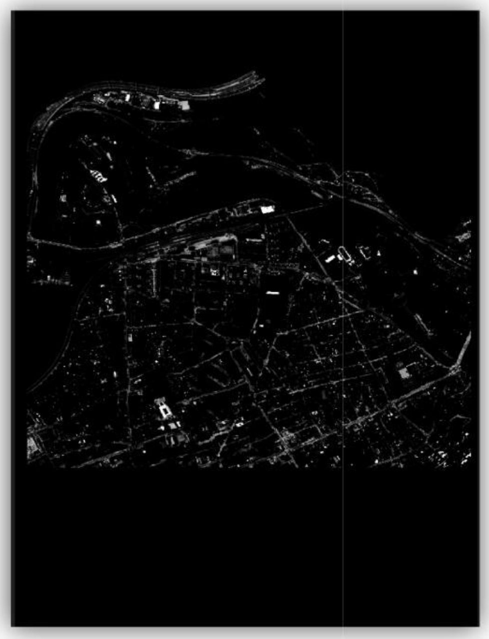
```
model.add(LSTM(2 ** 7,2, return_sequences="True", input_shape=[12, 1]))
model.add(MaxPooling1D(2))
model.add(LSTM(2**7,return_sequences=”True”)
model.add(MaxPooling1D(2))
model.add(LSTM(2**7))
model.add(Dense(n_classes,activation=’sigmoid’))

model.compile(loss=’catergorical_crossentropy’,optimizer=’adam’,metrics=[‘accuracy’]
model.summary()
model.fit(X,y_train,batch_size=10,epochs=1500)
y_test_new=model.predict(x_test,batch_size=50)
```

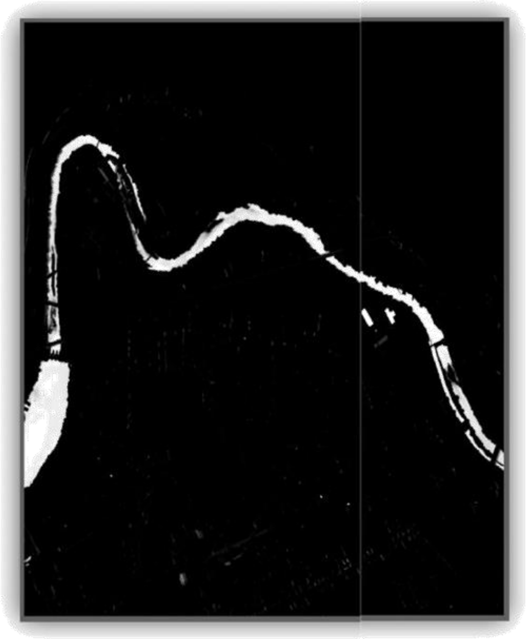
Run the model as many times as epochs is defined in batch size. Predict the model

Save images in specified folders : images are saved in .tiff format

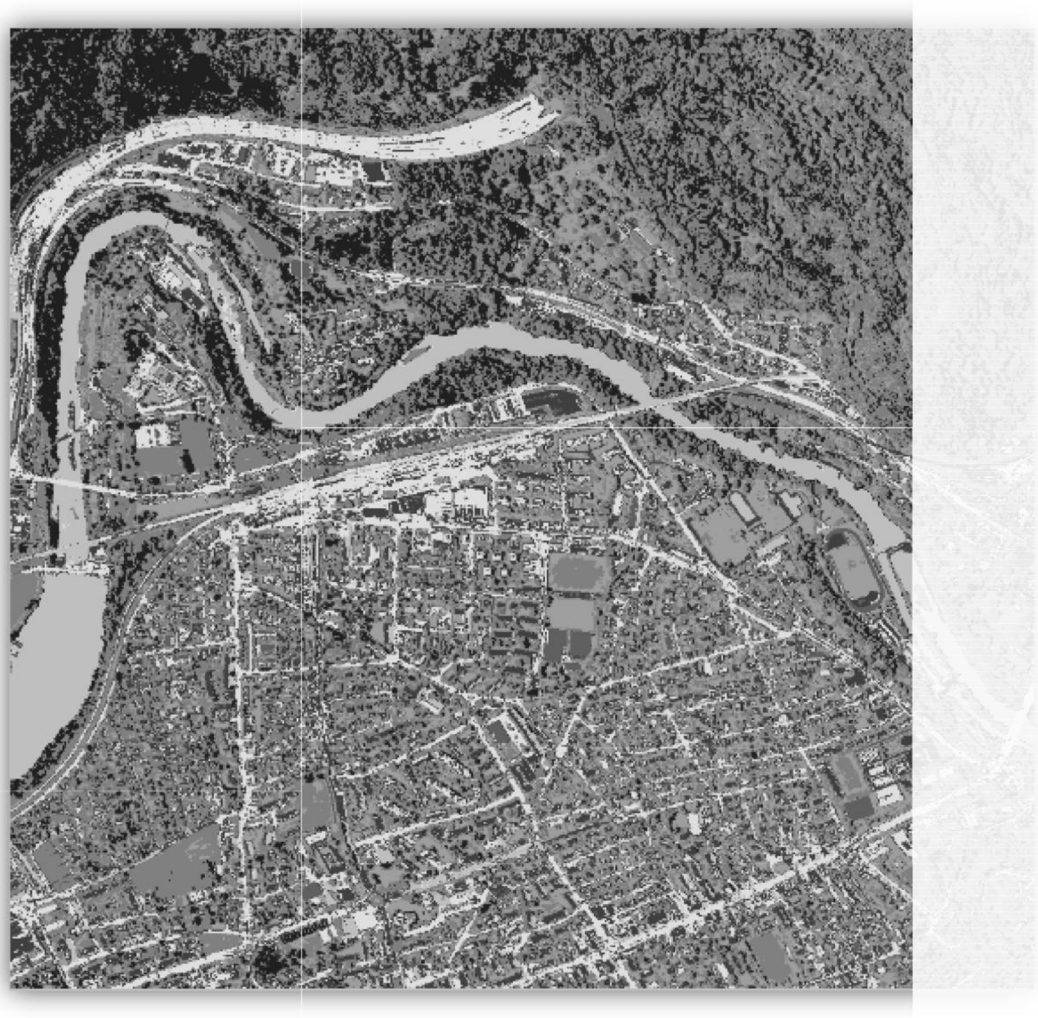
OUTPUT:

For Soft Classification:

 **Road class**	 **Dull-For est 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 i mage 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, grassland s 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.