Satellite Image Classification Summary Report

12/3/2020

1. Objective

The object of this capstone project is to build a neural network model to classify any user image.

2. Data Source

The original publication is:

Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert Dibiano, Manohar Karki and Ramakrishna Nemani, DeepSat - A Learning framework for Satellite Imagery, ACM SIGSPATIAL 2015.

(http://csc.lsu.edu/~saikat/deepsat/)

Kaggle CSV data file:

https://www.kaggle.com/arpandhatt/satellite-image-classification

3. EDA

SAT-4 consists of a total of 500,000 image patches covering four broad land cover classes. These include - barren land, trees, grassland and a class that consists of all land cover classes other than the above three. 400,000 patches (comprising of four-fifths of the total dataset) were chosen for training and the remaining 100,000 (one-fifths) were chosen as the testing dataset. We ensured that the training and test datasets belong to disjoint set of image tiles. Each image patch is size normalized to 28x28 pixels. Once generated, both the training and testing datasets were randomized using a pseudo-random number generator.

The MAT file for the SAT-4 dataset contains the following variables:

train_x 28x28x4x400000 uint8 (containing 400000 training samples of 28x28 images each with 4 channels)

train_y 400000x4 uint8 (containing 4x1 vectors having labels for the 400000 training samples)
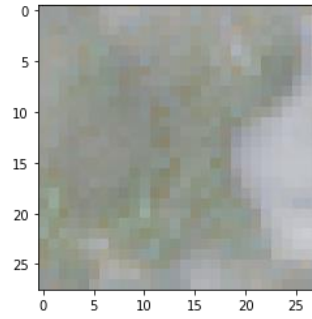
test_x   28x28x4x100000 uint8 (containing 100000 test samples of 28x28 images each with 4 channels)

test_y   100000x4 uint8 (containing 4x1 vectors having labels for the 100000 test samples)
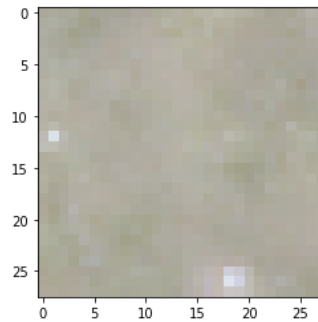

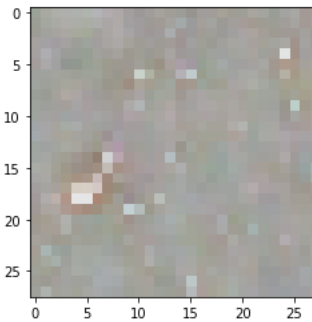There are four categories: barren land, trees, grassland, other
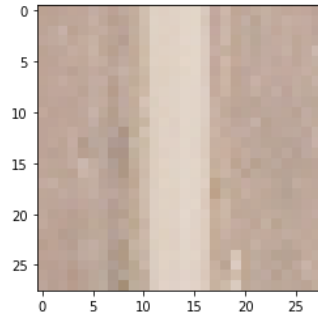
Barren land (label 0, 104465 images)
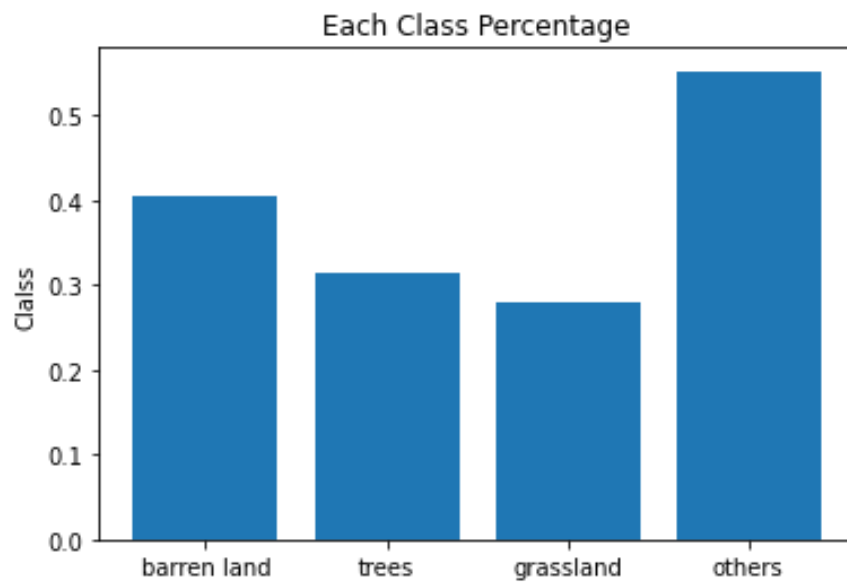

Trees (label 1, 81118 images)

Grassland (label 2, 72017 images)

Others (label 3, 142400 images)

The four categories are evenly distributed, database is balanced. There is no over sample or under sample issues.


4. Modeling

Used two models, general deep neural network and convolutional neural network.

4.1 deep neural network

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 10)	31370
dense_1 (Dense)	(None, 10)	110
dense_2 (Dense)	(None, 4)	44

Total params: 31,524
Trainable params: 31,524
Non-trainable params: 0

The model doesn't work very well, loss function is about 0.65 and accuracy is 0.72.
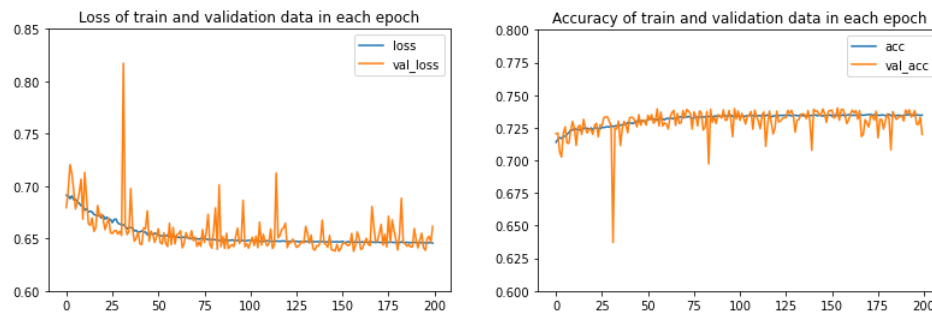
Model evaluation

```
Deep Neural Network: Accuracy=0.719
Deep Neural Network: f1-score=0.704
              precision    recall  f1-score   support

           0       0.84      0.92      0.88     26189
           1       0.68      0.84      0.75     20231
           2       0.56      0.31      0.40     17946
           3       0.70      0.70      0.70     35634

    accuracy                           0.72    100000
   macro avg       0.69      0.70      0.68    100000
weighted avg       0.71      0.72      0.70    100000
```

Confusion Matrix

4.2 Convolution neural network

Model architecture

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 10)	370
dropout (Dropout)	(None, 28, 28, 10)	0
conv2d_1 (Conv2D)	(None, 26, 26, 10)	910
flatten (Flatten)	(None, 6760)	0
dense_6 (Dense)	(None, 4)	27044

Total params: 28,324
Trainable params: 28,324
Non-trainable params: 0

Model performance

Model performed very well, the loss function decreased quickly and keep constantly around 0.1. The accuracy is 0.98
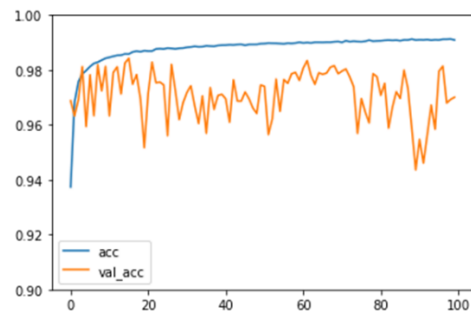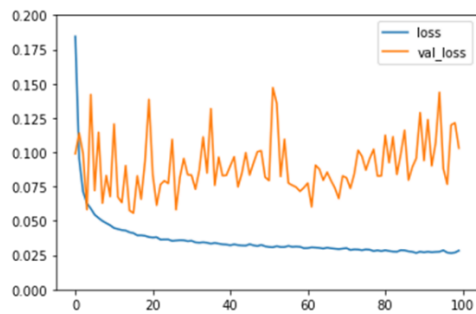
Model Evaluation

```
CNN: Accuracy=0.971
CNN: f1-score=0.704
              precision    recall  f1-score   support

           0       0.99      0.93      0.96     26189
           1       0.99      0.99      0.99     20231
           2       0.89      0.99      0.93     17946
           3       0.99      0.98      0.99     35634

    accuracy                           0.97    100000
   macro avg       0.97      0.97      0.97    100000
weighted avg       0.97      0.97      0.97    100000
```
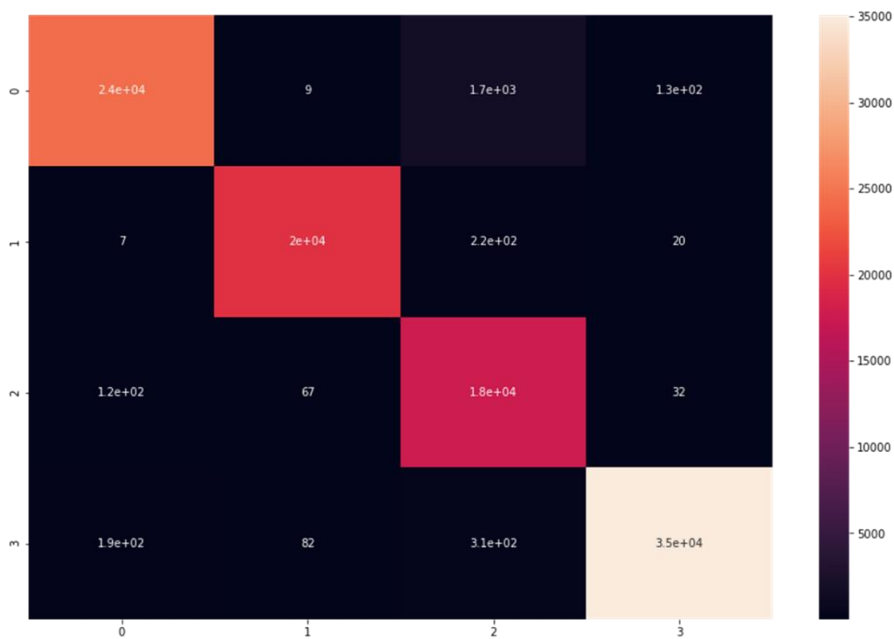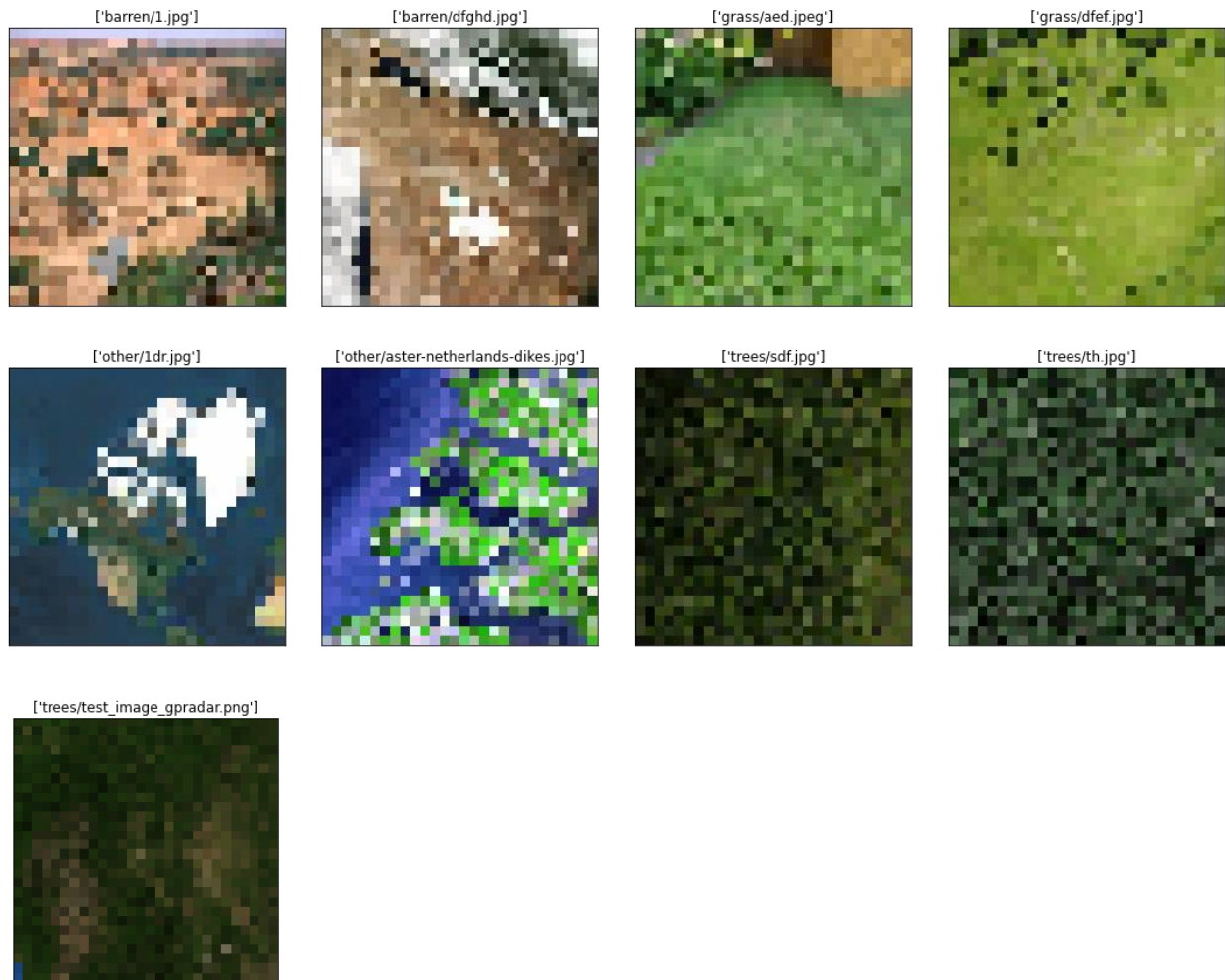
Confusion matrix

5. Testing

Download 9 satellite images from internet

Feed them into the CNN model and predict

Results: Predict all 9 images are others.

The problem here is all input data is 3 color channels, I artificially add 4th color channel as zero or mean of 3 channels. I think this cause the CNN model confused.

['barren/1.jpg']   ['barren/dfghd.jpg']   ['grass/aed.jpeg']   ['grass/dfef.jpg']

['other/1dr.jpg']   ['other/aster-netherlands-dikes.jpg']   ['trees/sdf.jpg']   ['trees/th.jpg']

['trees/test_image_gpradar.png']

Load the last image as 28*28*4 and predict with model, get grassland class, which is correct.
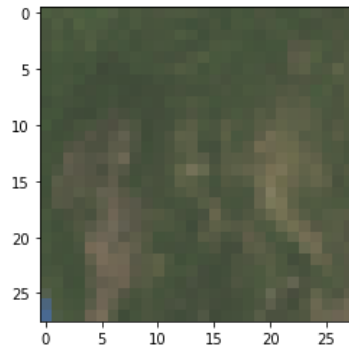
Image with 4 channels

6. Summary

Trained a large dataset using general deep neural network and CNN model, CNN is much better in terms of accuracy

Tested model on random pictures. It demonstrated the importance to have the infra color information. Samples need to be prepared and pre-processed to close to the training input to get more accurate results.

7. Future work

Label more features;

Train model only has 3 color channels;

Try U-Net to do object detection.

Test GANs model

8. Preliminary GANs results

GANs need more powerful computation, consider train the model on cloud or use large personal GPU.