Satellite Image Classification

Grace Yu

12/5/2020

Motivation

Build a machine learning framework to differentiate any satellite image by trained models

Highlights

➢ Different machine learning algorithms including deep neural network, CNN, and GANs models were deployed on a huge satellite image dataset.

➢ Model architecture and performance were explicitly evaluated.

➢ The best model, CNN model, was applied to classify new images successfully.

Data Sources

➢ Original Paper

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/

➢ Satellite Images CSV file
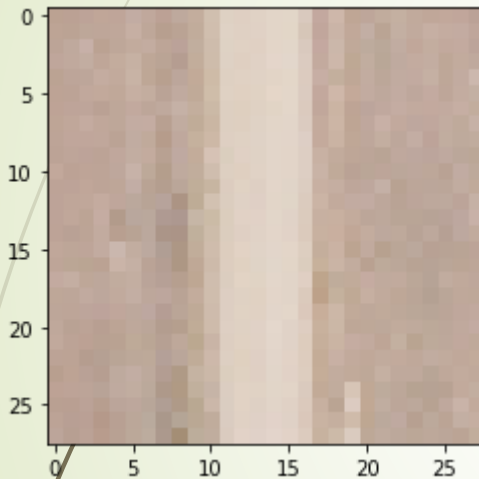
https://www.kaggle.com/arpandhatt/satellite-image-classification
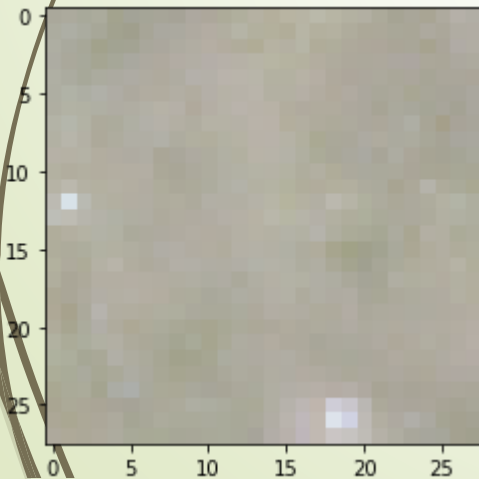
Exploratory Data Analysis

SAT-4 Image: 500,000, training: 400,000 (4/5), test: 100,000 (1/5)

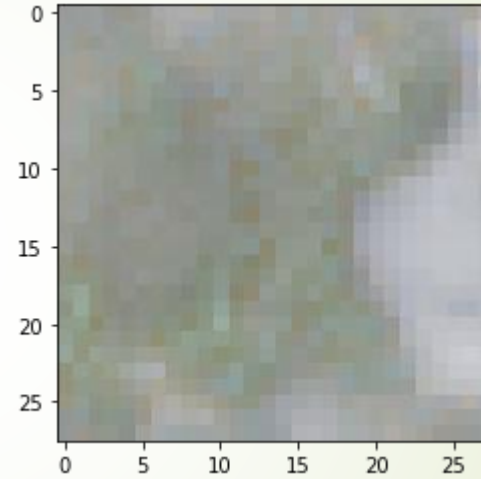Four categories: barren land, trees, grassland, other (in this order)

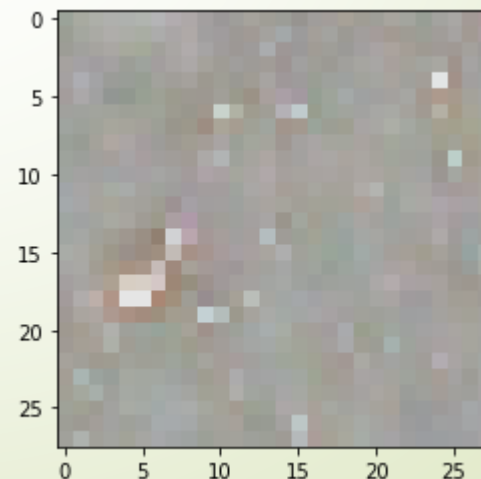Each image: 28x28x4 (RGB and Infra).

**Barren land
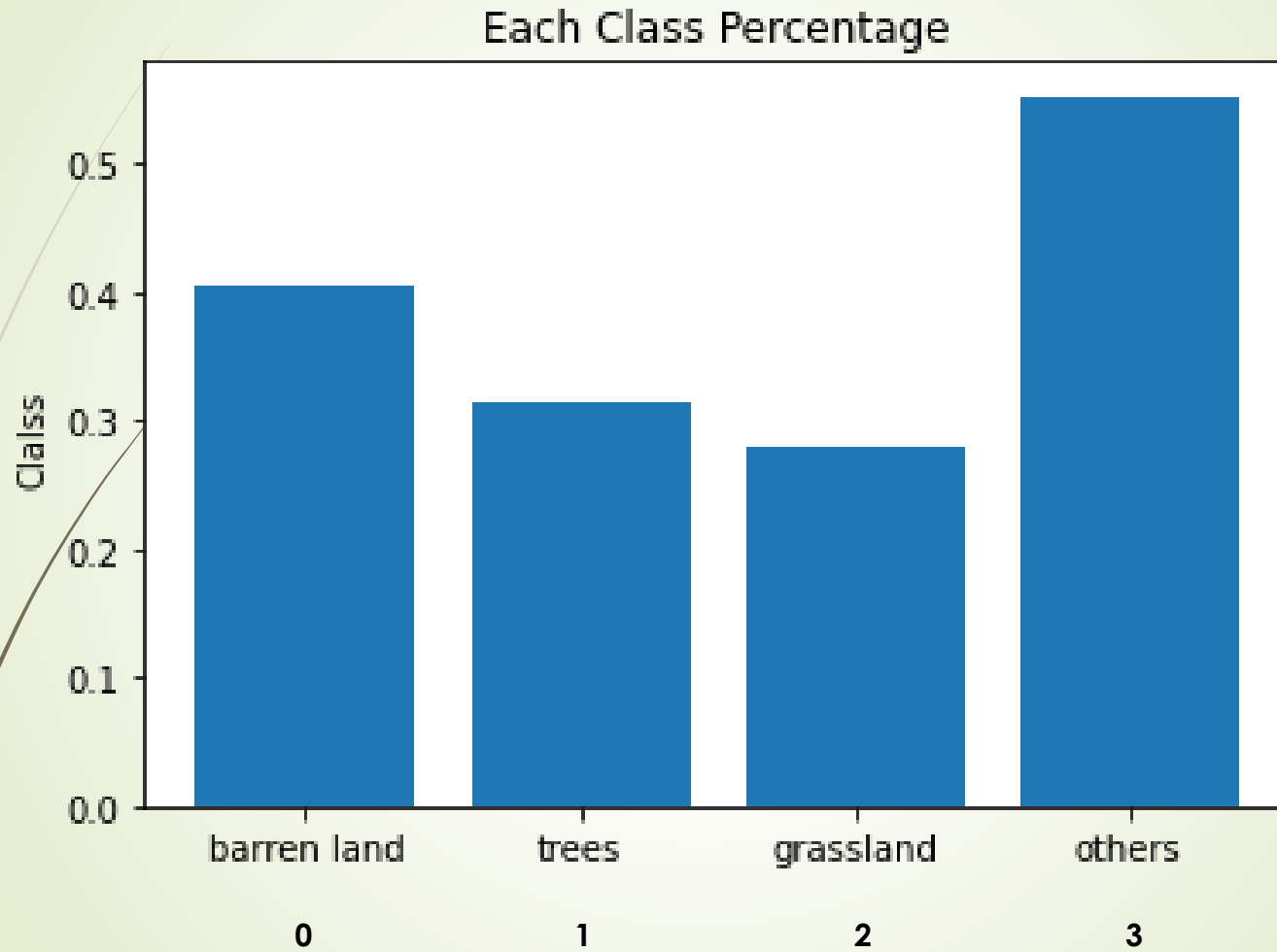(label: 0)
(104,465)**

**Trees
(label: 1)
(81,118)**

**Grassland
(lable: 2)
(72,017)**

**Other
(lable: 3)
(142,400)**

Samples are balanced

Model 1 - Deep Neural Network

Hidden
(2 layers)

Parameter = 3136*10+3136 = 34496

Parameter = 110

Parameter = 50

w1
w2
w3
w4
w5
w6
w7
w8
w9
w10

3136 = 28x28x4

10

10

4
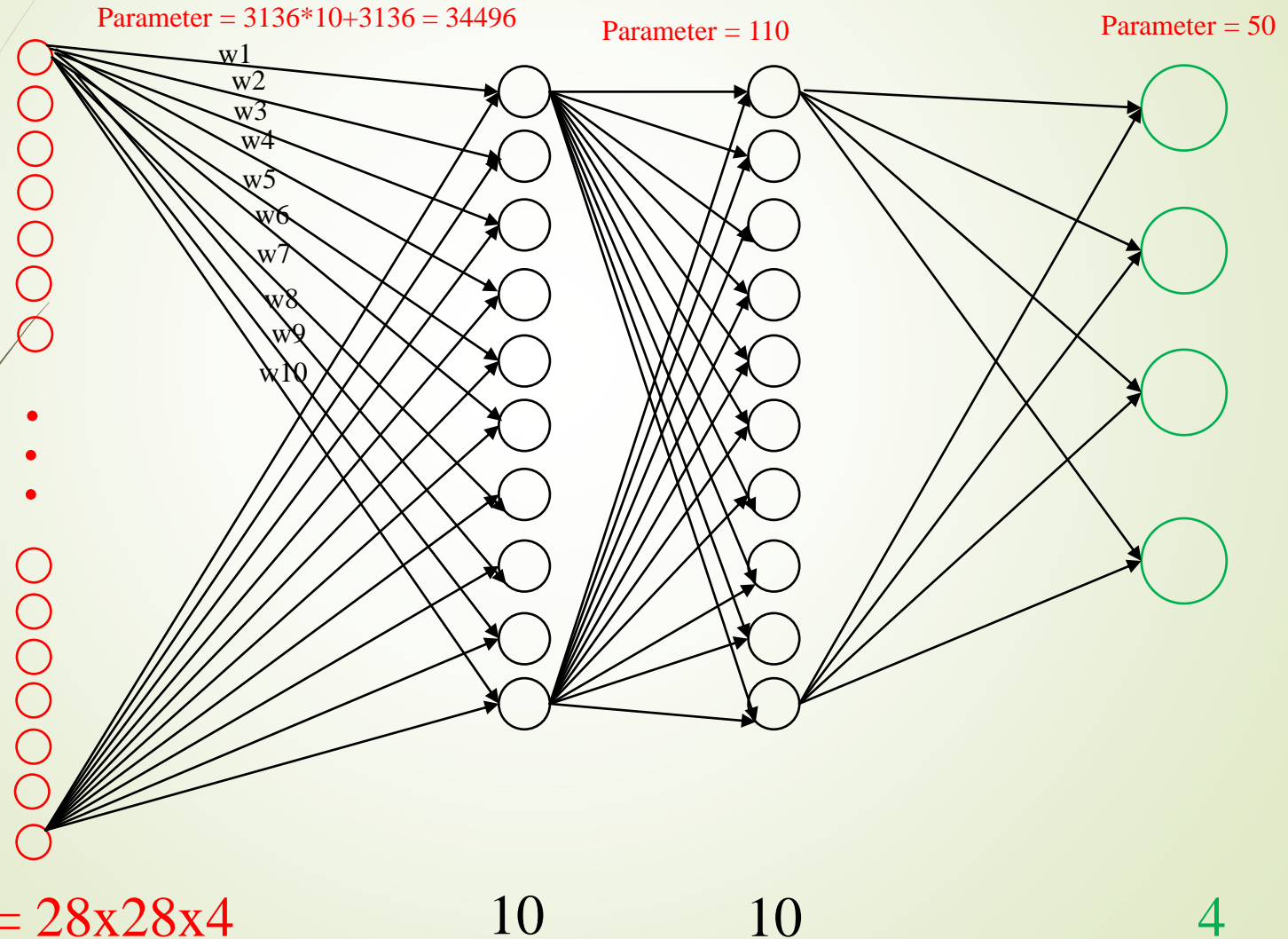
Model Architecture

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

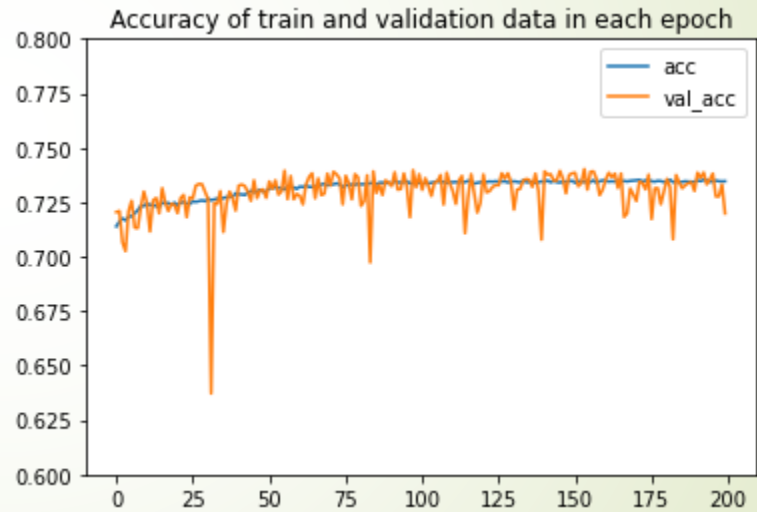# Model Performance

Training Data

Loss Function

Accuracy

Loss of train and validation data in each epoch
— loss
— val_loss

Accuracy of train and validation data in each epoch
— acc
— val_acc
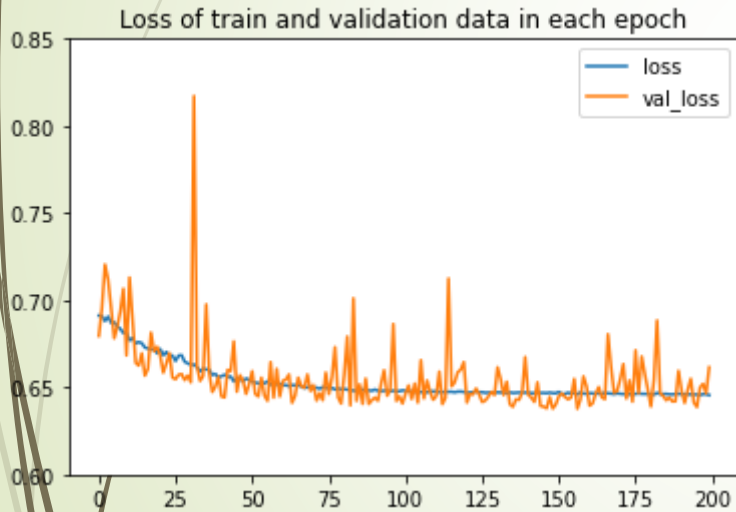
Model Evalation

```
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
```
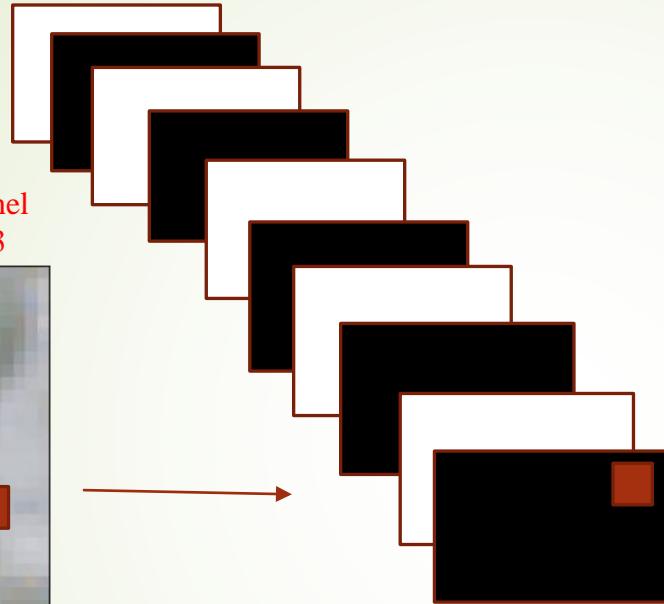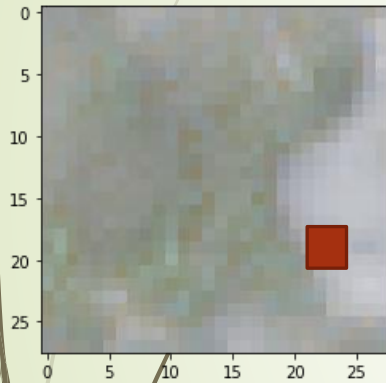
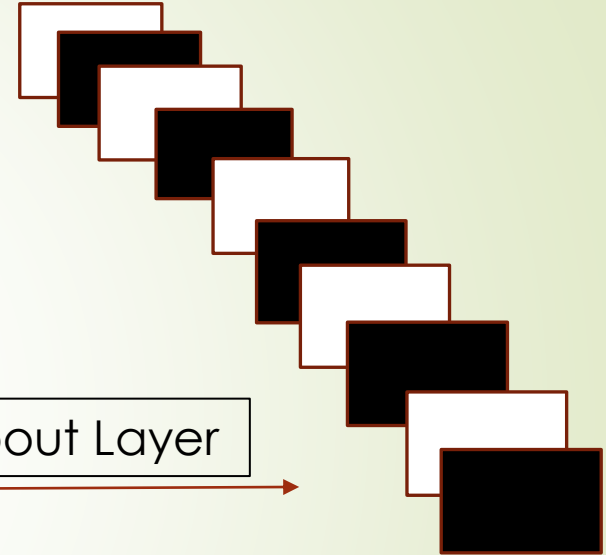# Confusion Matrix

Model 2 – Convolutional Neural Network (CNN)

10@28x28

10@26x26

Kernel
3x3

Dropout Layer

Fully
Connected
Layer

Flatten

4 Output

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

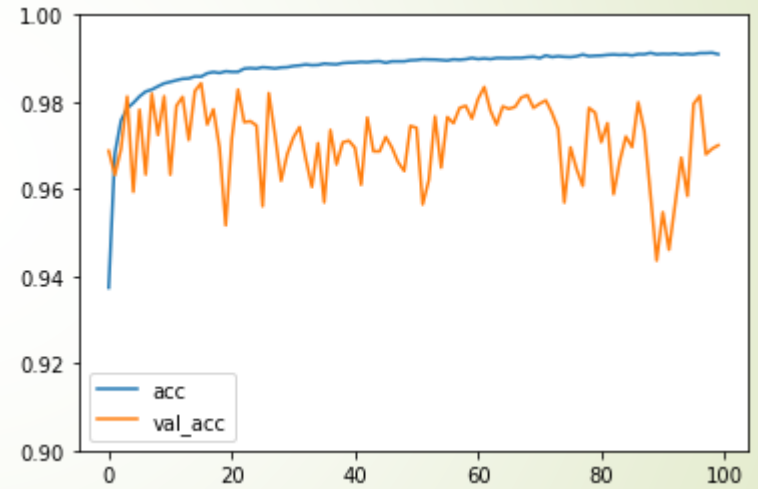Total params: 28,324
Trainable params: 28,324
Non-trainable params: 0

Model Performance
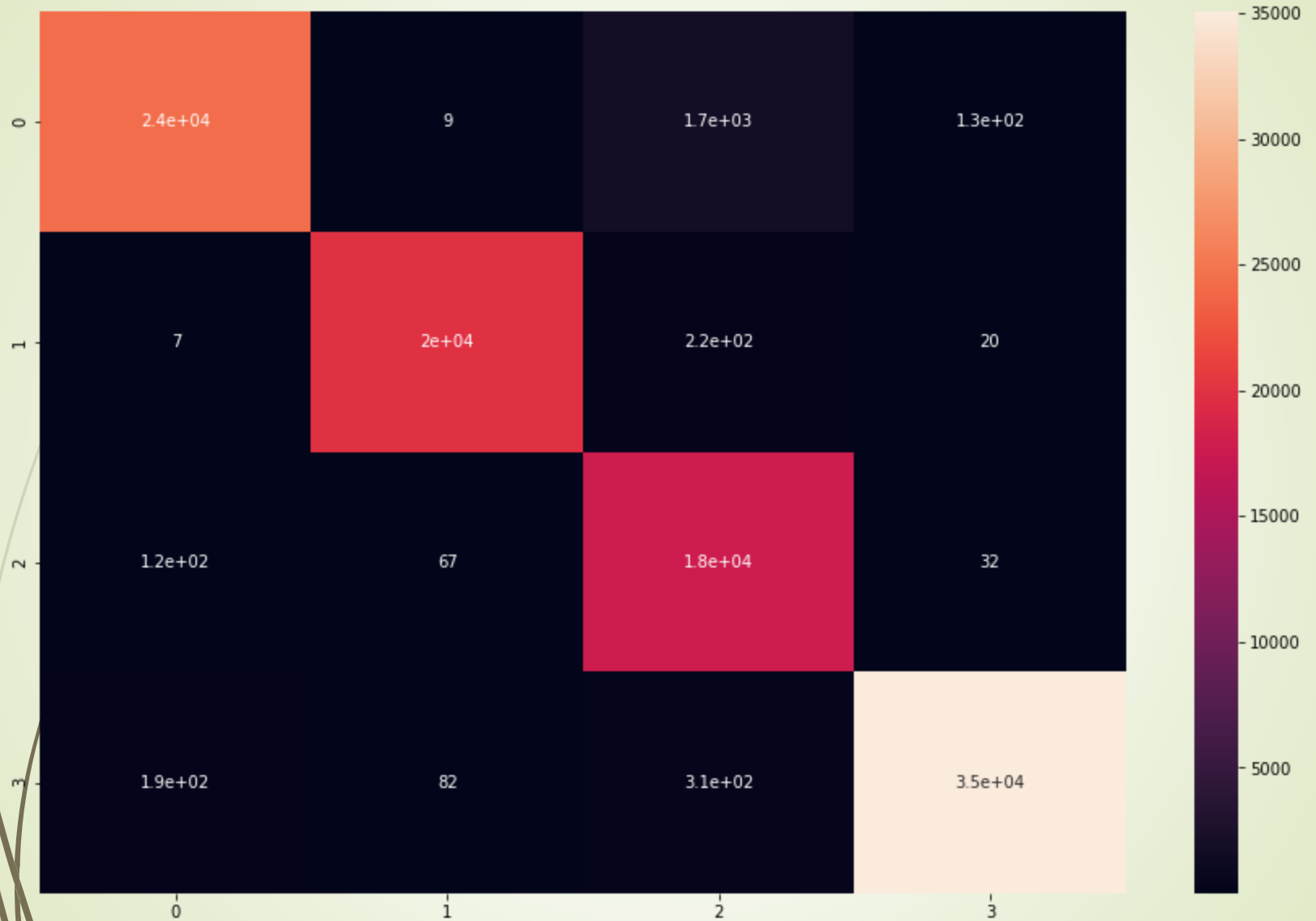
Training Data

Loss Function
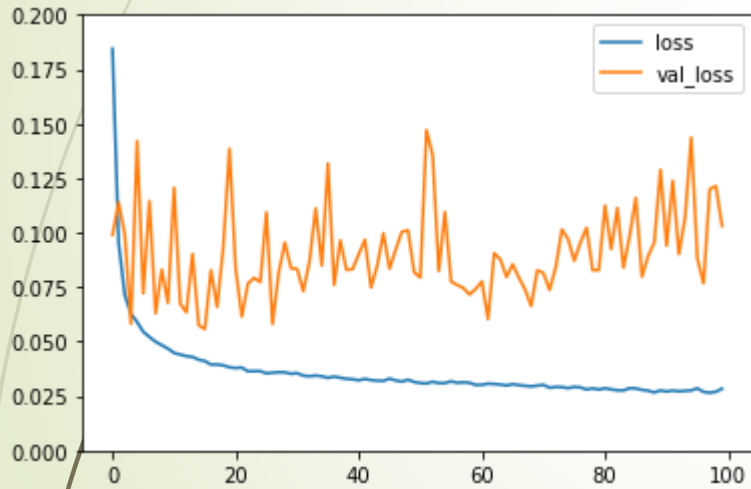
Accuracy

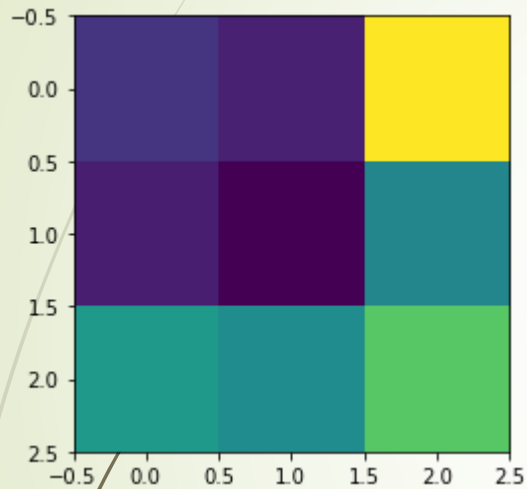# Model Evalation

```
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

Model Interpretation
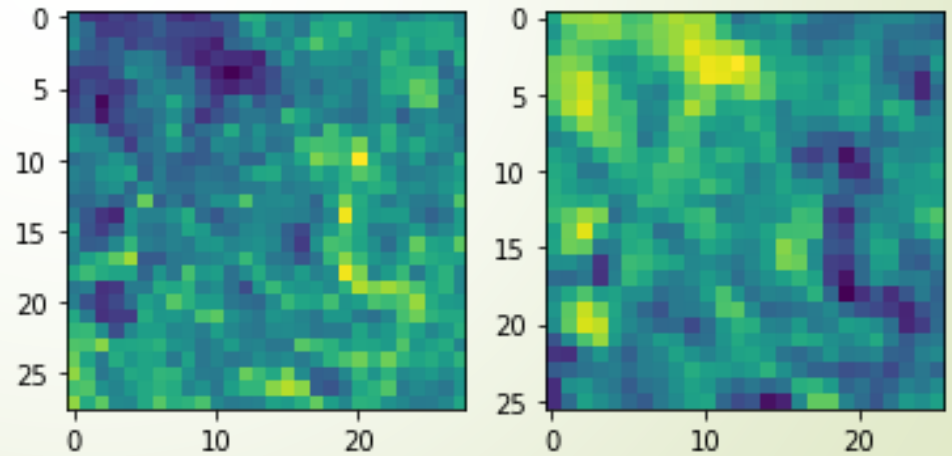
Kernel – Layer1

Image
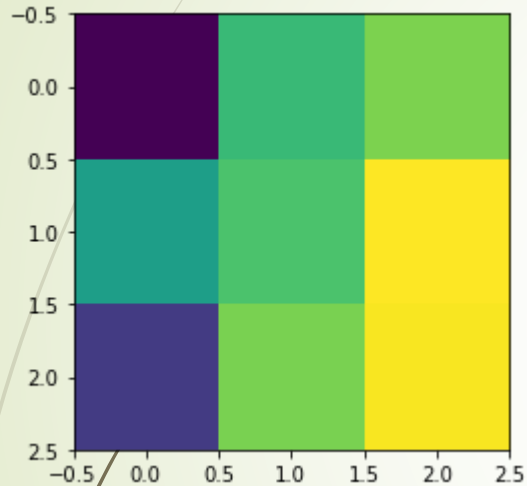
Before   After

Model Interpretation

Kernel – Layer2
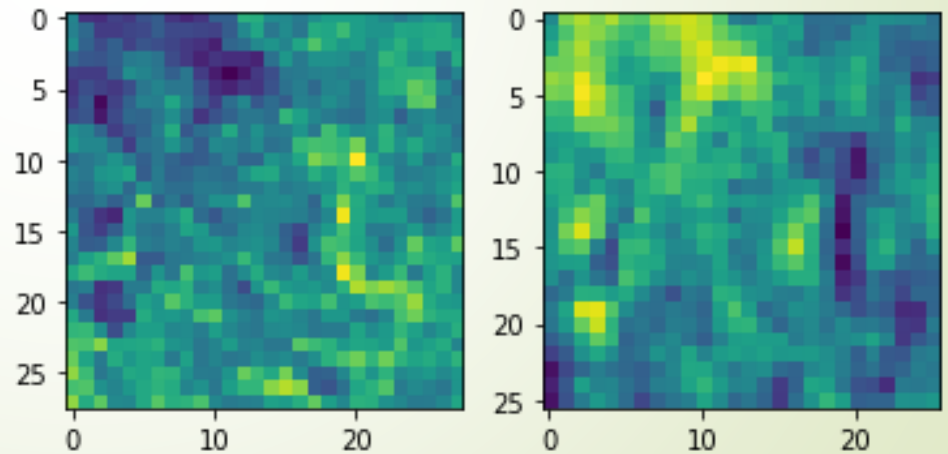

Image

Before   After

Image for Testing

['barren/1.jpg']

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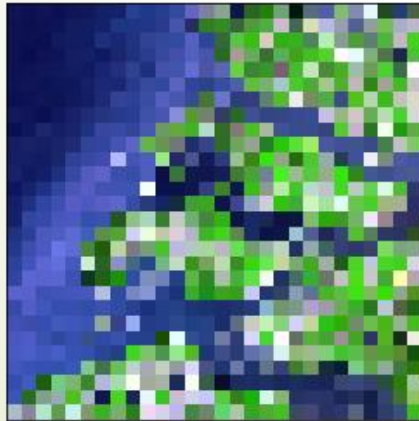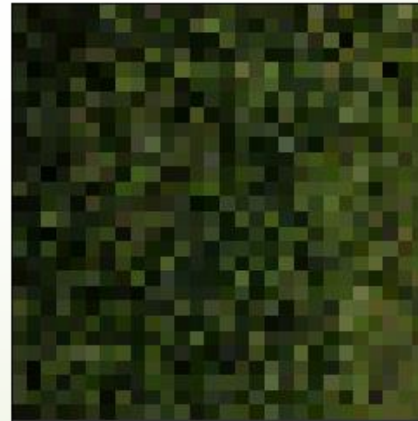
['grass/aed.jpeg']

['grass/dfef.jpg']

['other/1dr.jpg']

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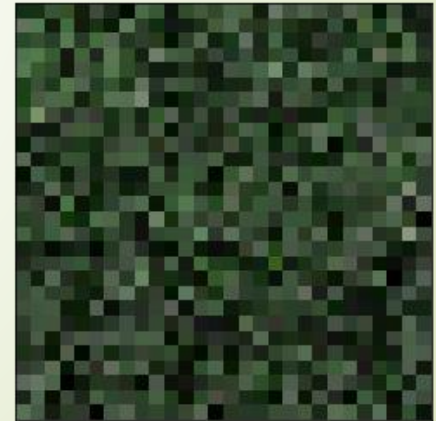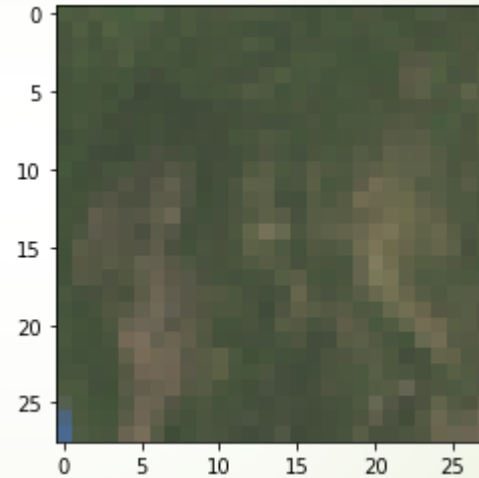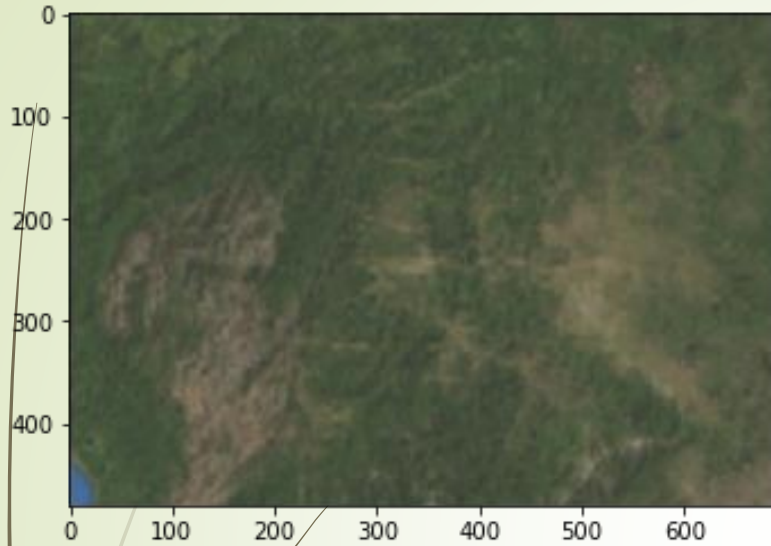
['trees/sdf.jpg']

['trees/th.jpg']

Image Prediction

- 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.
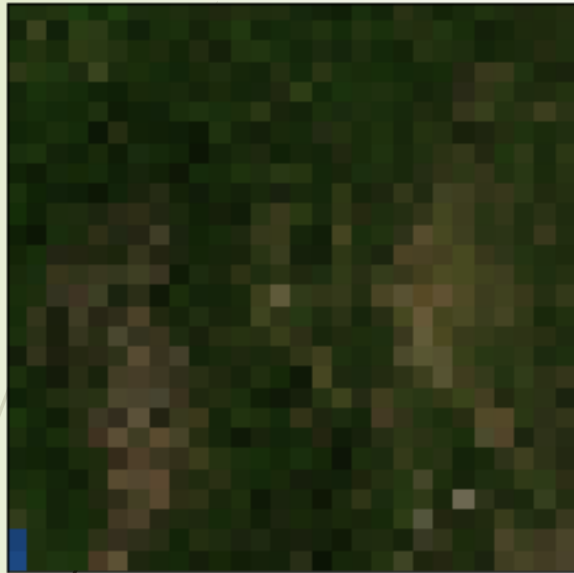
Results

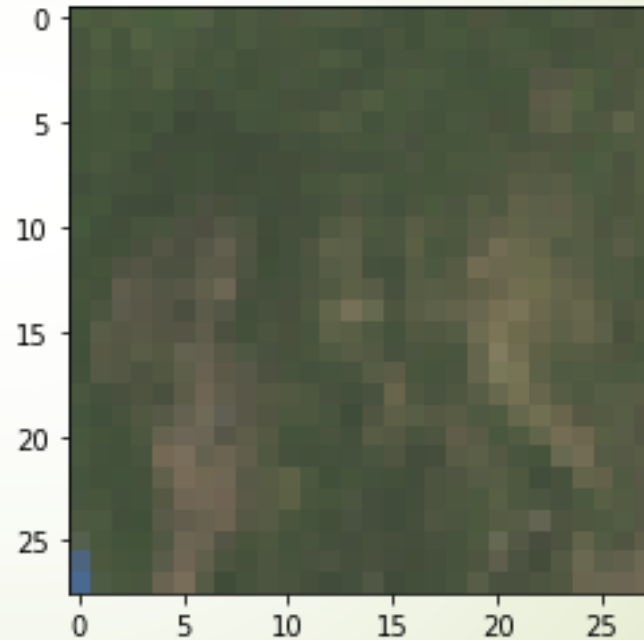## Prediction: 2 - Grassland

Discussion

3 channels

['trees/test_image_gpradar.png']

Prediction: 3 – Other

Information loss

4 channels

Prediction: 2 – Grassland

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 match the training input for more accurate prediction.**
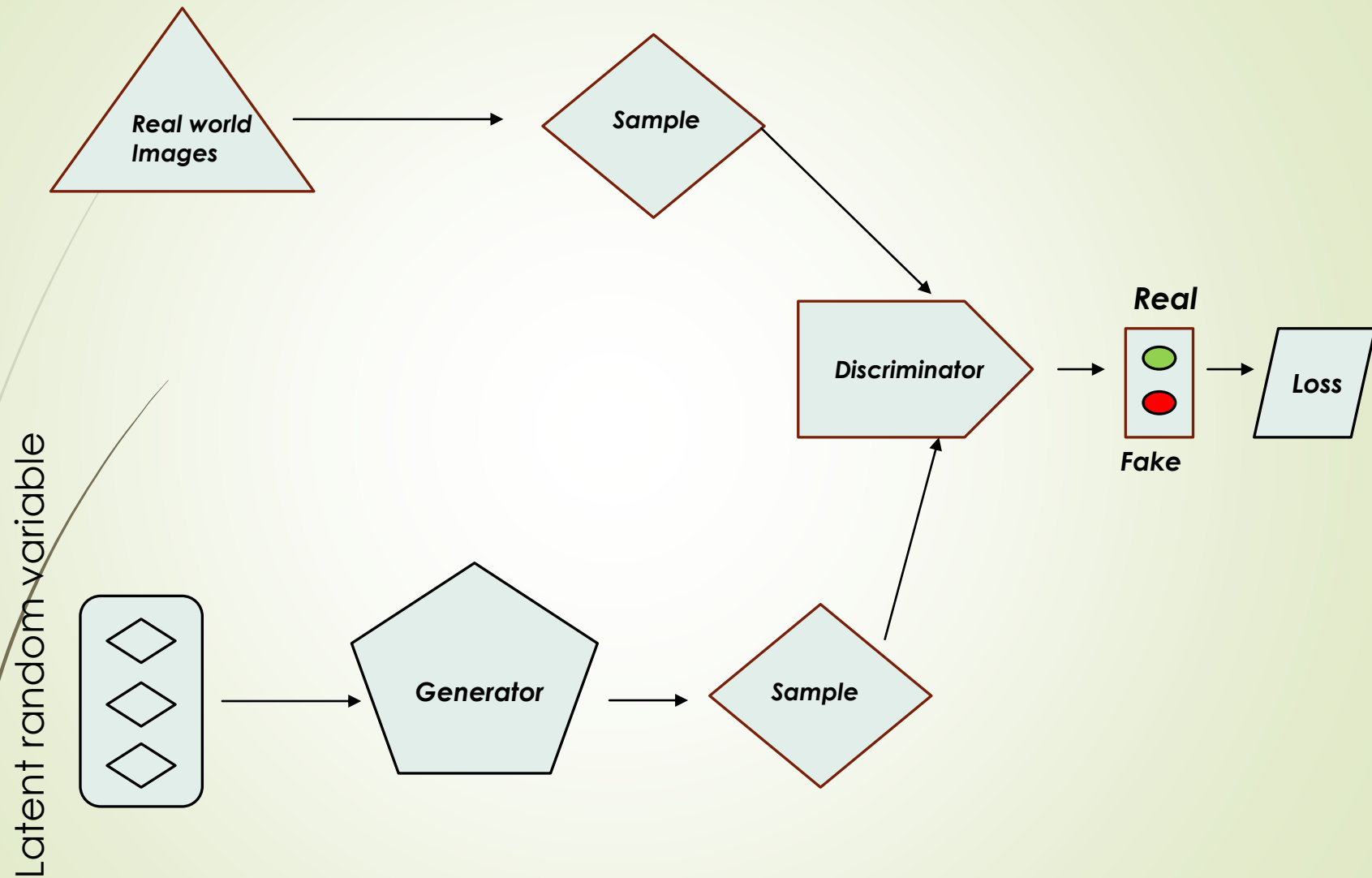
Future Work

➢ **Label more features, can train SAT-6 data;**

➢ **Train model with 3 color channels only;**

➢ **Use satellite image with more channels to train classification model;**

➢ **Try U-Net to do object detection;**

➢ **Test GANs model to generate synthetic images for training.**

Preliminary GANs Results

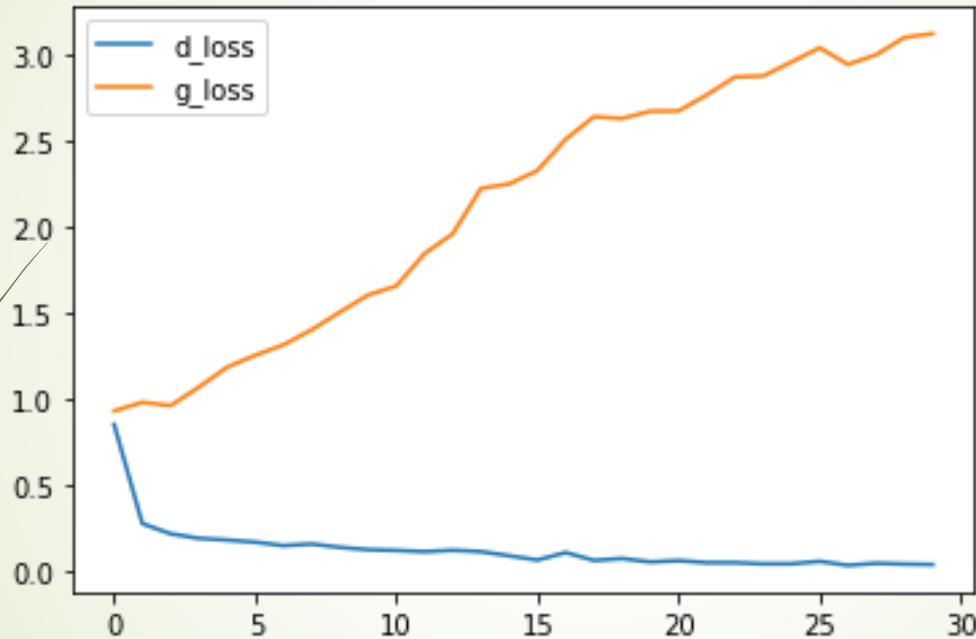# Generative Adversarial Networks (GANs)

Model Architecture

```
Model: "functional_11"
_____
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         [(None, 100)]             0
_____
functional_9 (Functional)    (None, 3136)              3321920
_____
functional_7 (Functional)    (None, 1)                 1737729
=================================================================
Total params: 5,059,649
Trainable params: 3,319,872
Non-trainable params: 1,739,777
_____
```
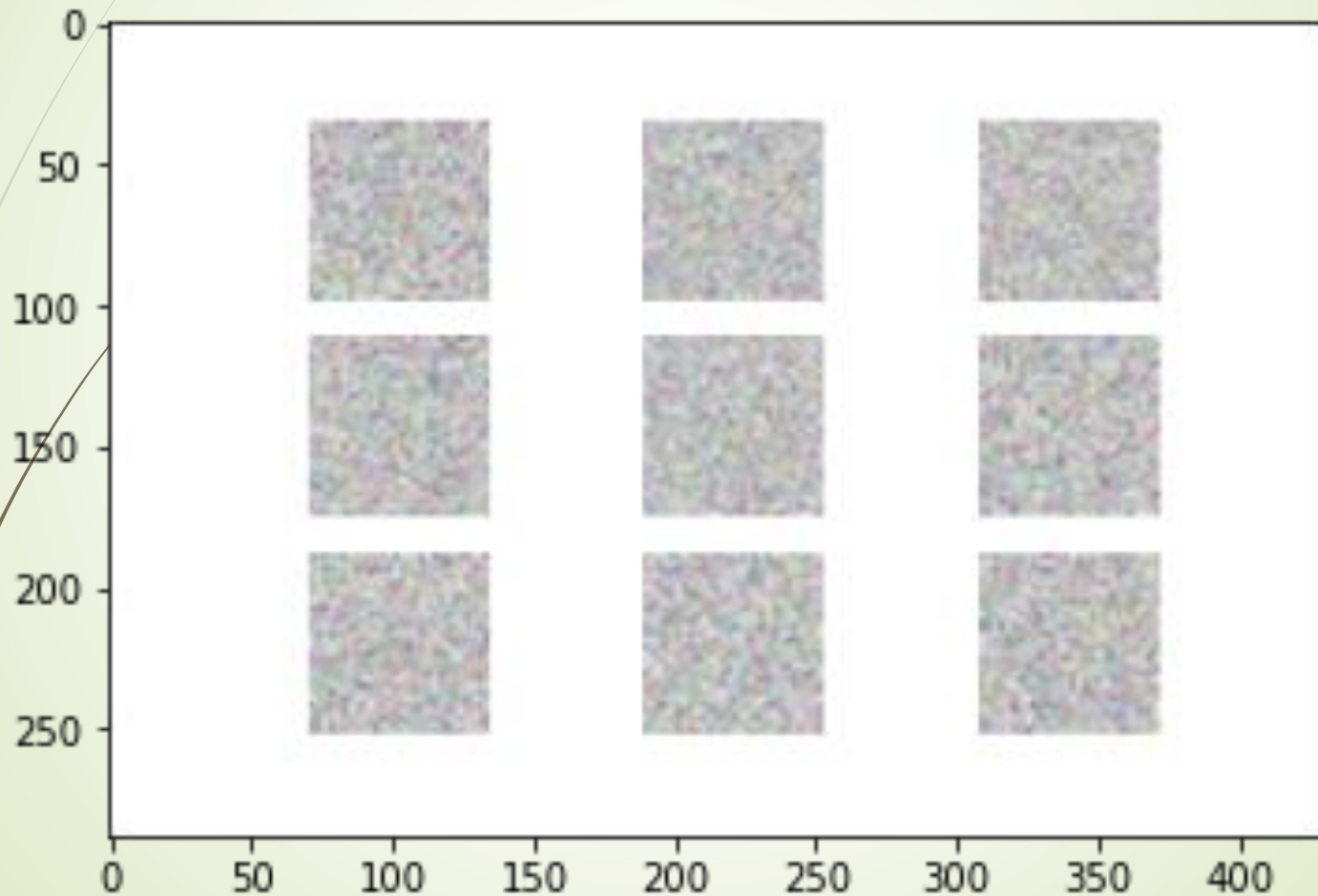
Model Performance

Loss Function

Discriminator converge very quickly, while generator is far from converging. Only run 300 epochs, need more epochs. Need cloud computation or GPU.
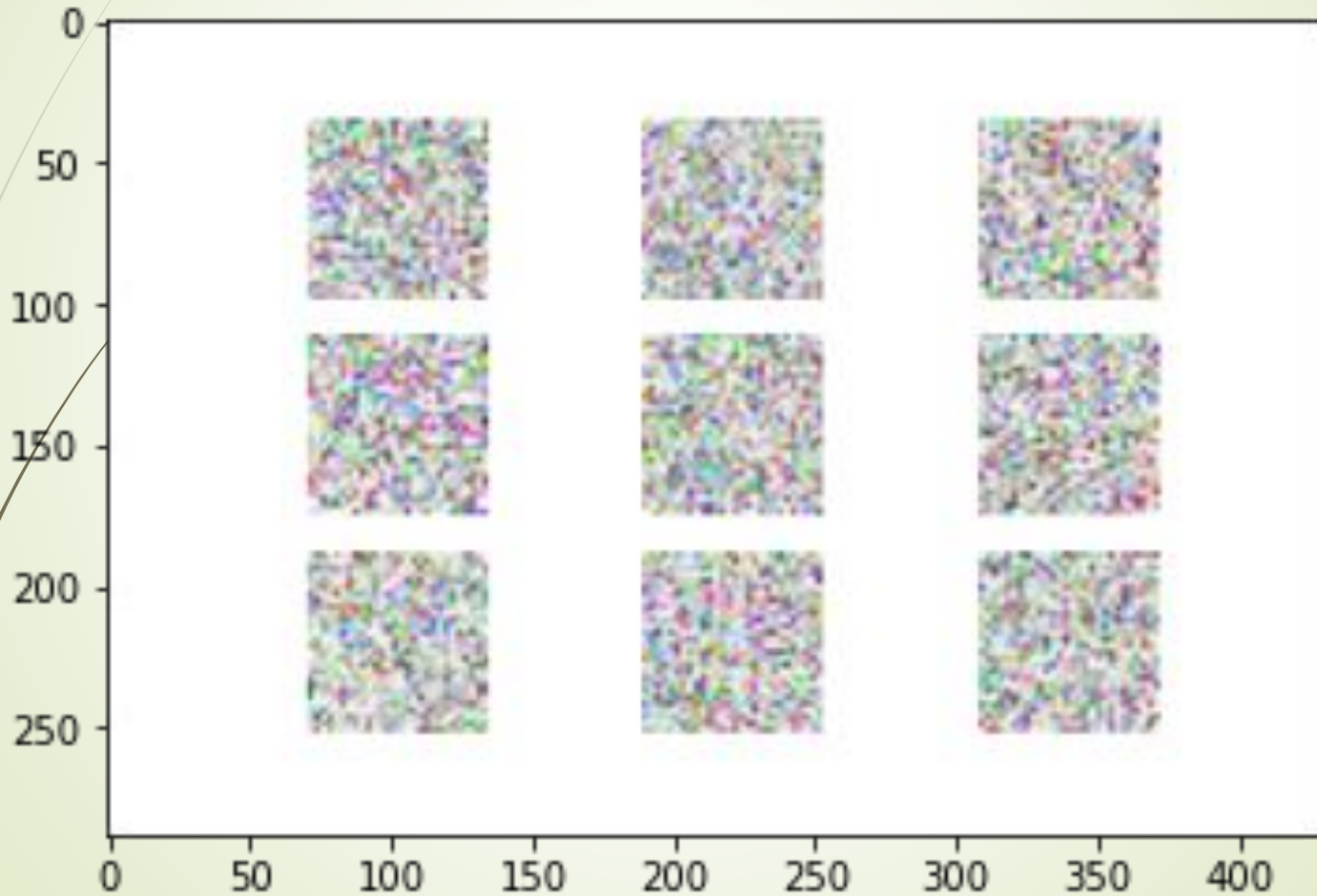
Synthetic Images

Beginning

Synthetic Images
Middle

Synthetic Images
End