Multispectral Satellite Image Classification with VGG16

Alan Mai
George Hong
Gryffan Palmer
Anthony Ayon

Motivation

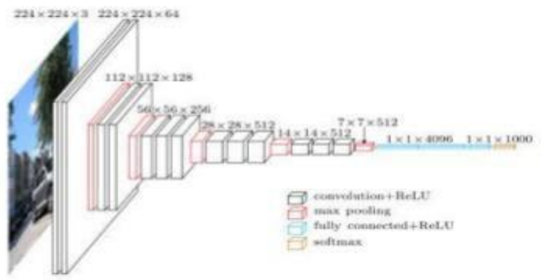
- Learn how to use newly developed deep neural networks
- Classify satellite images based on their terrain
- Experiment with transfer learning
  - Understand how well VGG16 pretrained with ImageNet transfer learns to satellite images
- Investigate ways to incorporate multispectral imagery with VGG-16
- Satellite imagery important for analysis of environmental development and degradation.

Anthony Ayon

What is VGG16?

VGG16 is a convolutional neural network (CNN) developed and named after the Oxford Visual Geometry Group. It was the runner up for ImageNet Classification in ILSVRC 2014, & uses the idea of shrinking spatial dimensions, but growing its depth. Keras has its own implementation of the VGG16() model and this is the one we used for our project.

VGG16 Pre-Trained Model



The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

Anthony Ayon

Citation: https://arxiv.org/pdf/1409.1556.pdf

Related Coursework

Neural Networks:

- Represents a multilayer system where the output stands for the next input.
- CNN is a specific type of neural network, mainly used for image processing.
- **Theory** should be similar; processors working linearly until it reaches its last layer, hence, the output
- Core difference with CNN is its tendency for layers to decrease/shrink the deeper they become
- In CNN, the input goes through a series of convolution and pooling(optional) layers.

Related Papers

Benchmarking Deep Learning Frameworks for the Classification of Very High Resolution Satellite Multispectral Data (M. Papadomanolakia, M. Vakalopouloua, S. Zagoruykob, K. Karantzalos)

- Used AlexNet variations and VGG to classify SAT-4 and SAT-6 data
- Application of VGG along with accuracies which led to accuracy >**99.98%**
- Used as a benchmark for our results

Multi-label Classification of Satellite Images with Deep Learning(D. Gardner, D. Nichols)

- Dataset consisted of satellite images of Amazon rainforest
- Incorporated VGG16 to classify particular changes inside rainforest, such as illegal logging activities
- F2 calculated for each model used, VGG16 was beaten by ResNet-50

Anthony Ayon

The SAT-4 Dataset

We used the SAT-4 dataset which has satellite images of different types of terrain(rural areas, urban areas, densely forested, mountainous terrain, small to large water bodies, agricultural areas, etc) covering California. Each sample is 28 pixels by 28 pixels.

Dimensions	28x28x4 (R,G,B,NIR)
Training Set	400,000
Testing Set	100,000
Complete Set	500,000

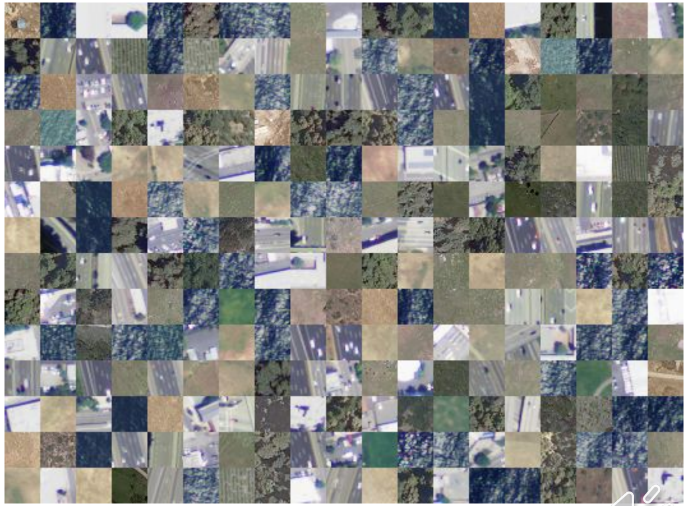
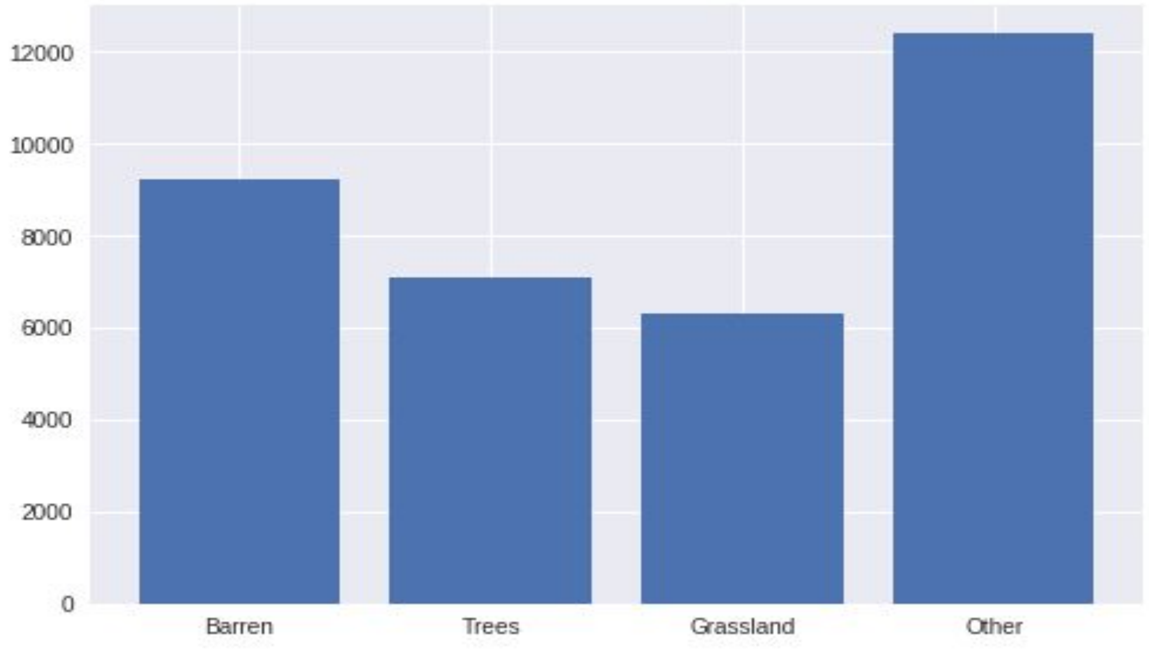


Image borrowed from https://csc.lsu.edu/~saikat/deepsat/

Gryffan Palmer

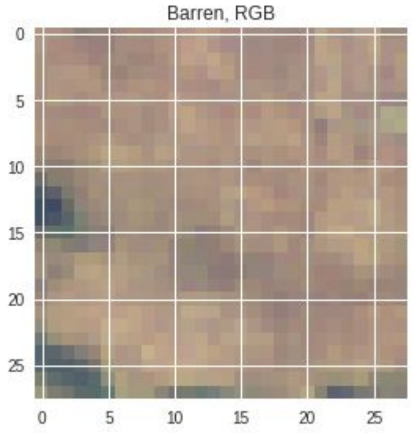
Number of Samples in Each Class

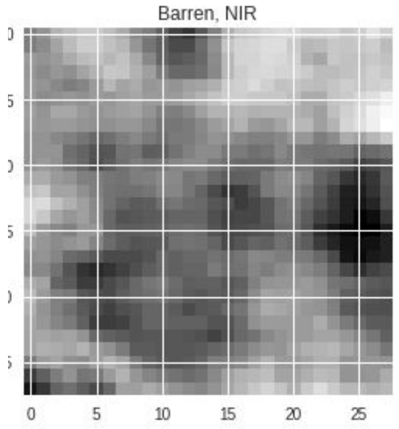


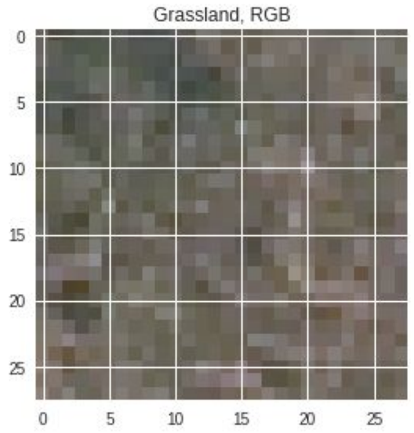
Notice:

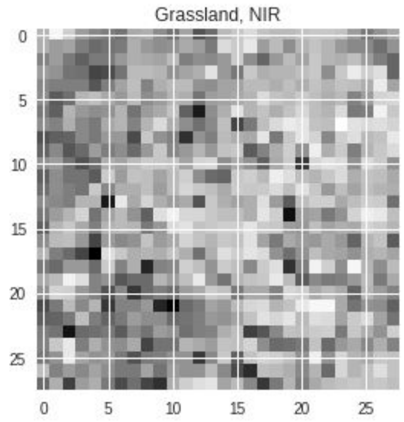
“Other/None” dominates over the other 3 classes since it emcompases multiple types of land types

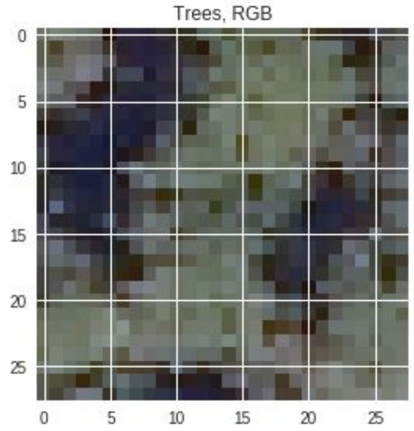
Gryffan Palmer

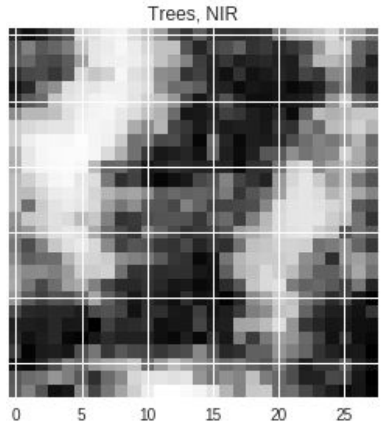


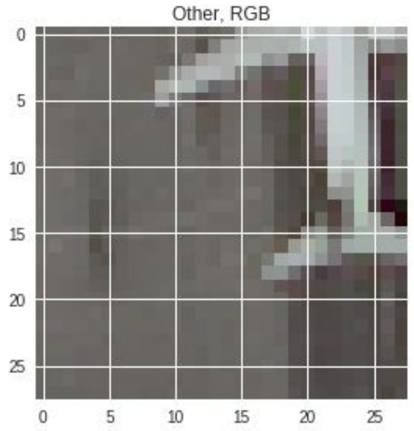


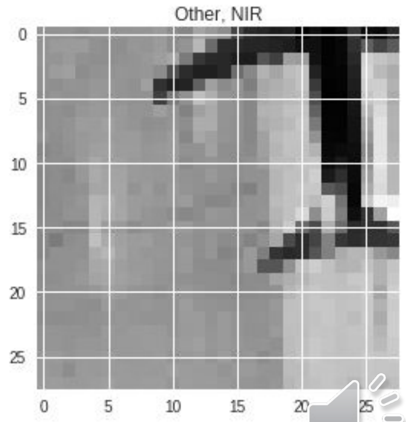












Gryffan Palmer

Images borrowed from the dataset

Preprocessing

Data (from Kaggle) had RGB-NIR in one long array
(number training samples x 3136)

Want: 3 channel image inputs with size ≥ 32 x 32 (minimum for keras-vgg16)

Preprocessing steps:

1. Reshape each image to RGB-NIR in separate channels (28 x 28 x 4)
2. Expand each image (without interpolation) to (56 x 56 x 4)
3. Separate the RGB (56 x 56 x 3) and NIR (56 x 56 x 1) channels
4. Duplicate the NIR channel across 3 channels (56 x 56 x 3)
5. Rescale the image values from [0,255] to [-1,1]

Gryffan Palmer

Alan Mai

Approach: VGG16 setup

- Split VGG16 into 3 parts:
  - Input, convolutional layers, fully connected layers
- Input modified for (56, 56, 3)
- Initialized convolutional layers with weights trained on ImageNet
  - Froze these layers to prevent training from modifying the weights
- Froze the convolutional layers with the weights trained on ImageNet.
- Added 2 fully connected with "relu" activation function below conv. layers, with 0.5 dropout after each
  - same FC and dropout setup as original VGG paper [10]
  - FC weights updated when training
- Add last 4 class "softmax" FC layer at very end
- loss = categorical cross entropy, optimizer = Adam with lr = 0.0001
- Loaded in our dataset in chunks due to limited memory capacity.

Alan Mai

Using Multispectral Images

- Combine the imagery from RGB and NIR
- DeepFruits [10] paper outlined two approaches
- Early Fusion: Alter the input of VGG to accept 4 channels rather than standard 3
    - Not implemented
- Late Fusion: Combine the classification information from models trained on RGB and NIR separately
    - `score[c] = mean(score_rgb[c], score_nir[c])`
        - `score[c] =` probability the sample is in class c
        - Used equal weighting between RGB and NIR
    - Requires more memory and time; must train two models.
- Compared VGG trained on RGB, NIR, and the results of late fusion of the two models

Experiment’s Results

Changing Batch Size :
Tested effect of different batch sizes on an **Adam** optimizer with a **0.0001 learning rate** with 4 epochs.

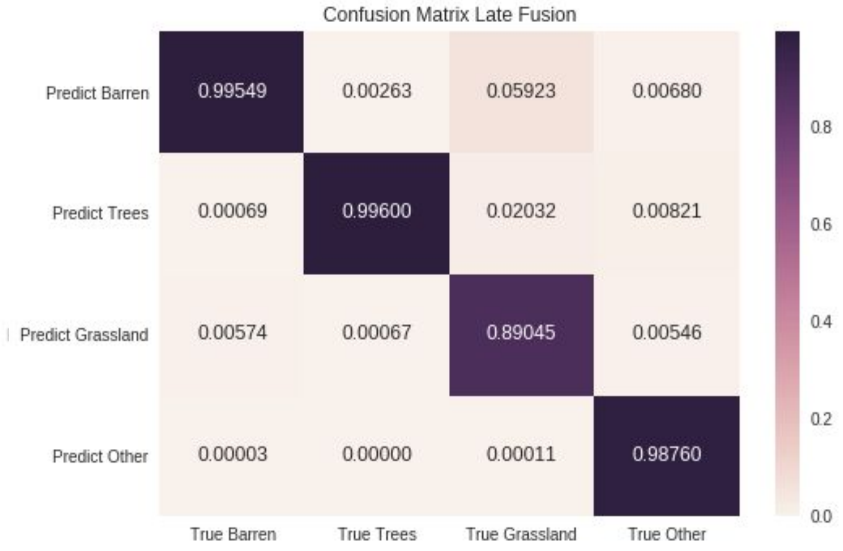
Batch Size	Training Time	Validation Accuracy	Loss
256	57min 32sec	98.618	0.03948
512	49min 16sec	98.371	0.04595
1024	55min 23sec	98.228	0.05251

George

Accuracy and F-score

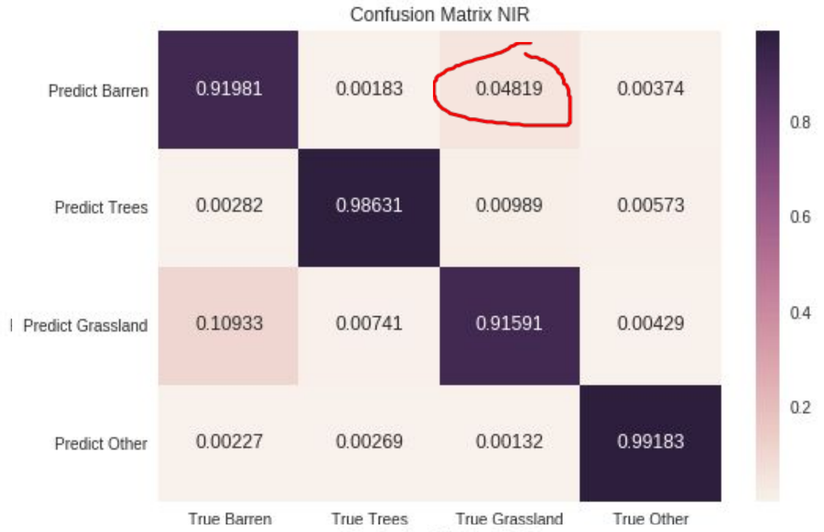
Model	**Accuracy**	**F1-score**
RGB	**98.785%**	**0.9858**
NIR	95.823%	0.9520
Late Fusion RGB+NIR	97.393%	0.9695

George



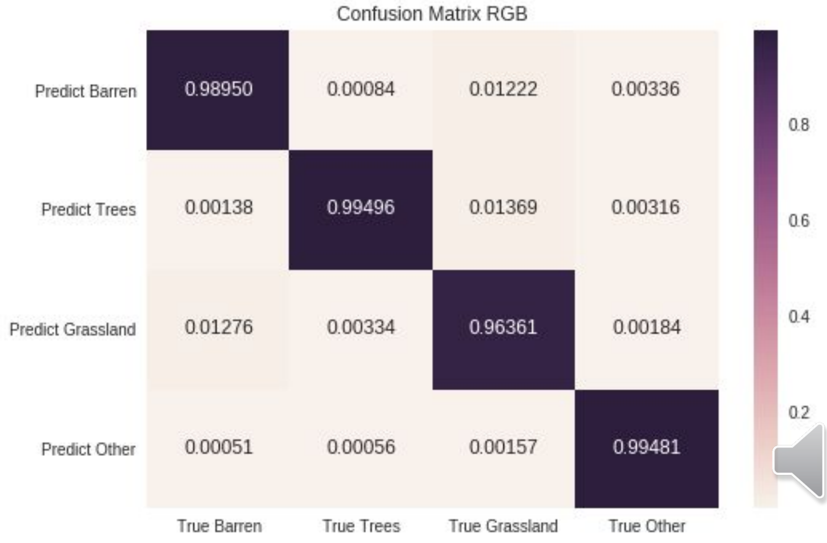
Confusion Matrix Late Fusion

	True Barren	True Trees	True Grassland	True Other
Predict Barren	0.99549	0.00263	0.05923	0.00680
Predict Trees	0.00069	0.99600	0.02032	0.00821
Predict Grassland	0.00574	0.00067	0.89045	0.00546
Predict Other	0.00003	0.00000	0.00011	0.98760



Confusion Matrix NIR

	True Barren	True Trees	True Grassland	True Other
Predict Barren	0.91981	0.00183	0.04819	0.00374
Predict Trees	0.00282	0.98631	0.00989	0.00573
Predict Grassland	0.10933	0.00741	0.91591	0.00429
Predict Other	0.00227	0.00269	0.00132	0.99183



Confusion Matrix RGB

	True Barren	True Trees	True Grassland	True Other
Predict Barren	0.98950	0.00084	0.01222	0.00336
Predict Trees	0.00138	0.99496	0.01369	0.00316
Predict Grassland	0.01276	0.00334	0.96361	0.00184
Predict Other	0.00051	0.00056	0.00157	0.99481

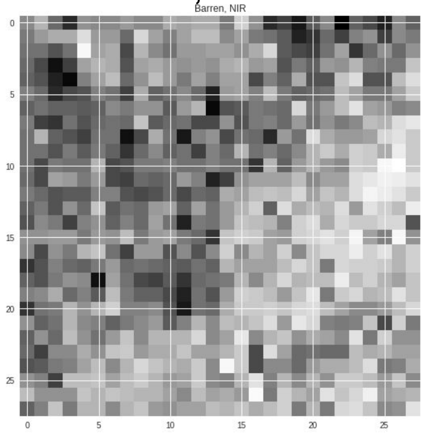
High Accuracy Overall.

NIR has slight trouble with Barren vs Grassland

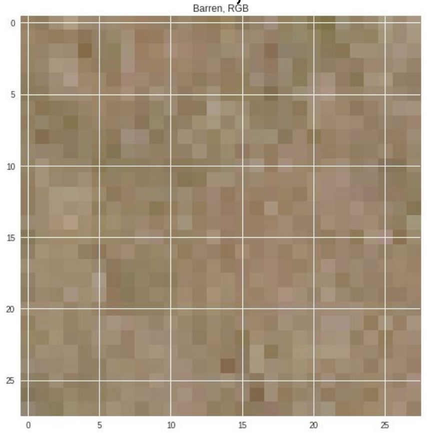
Late Fusion has best accuracy with Barren and Trees, and similar with Other but the grassland issue carried over from NIR.

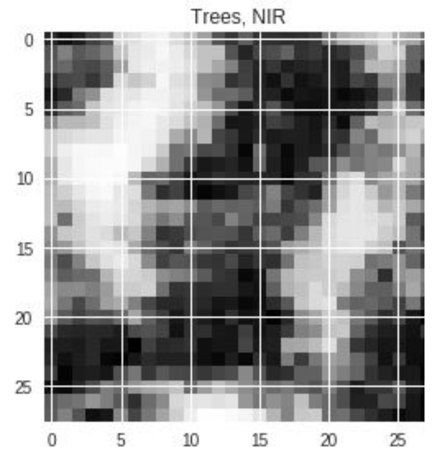
George

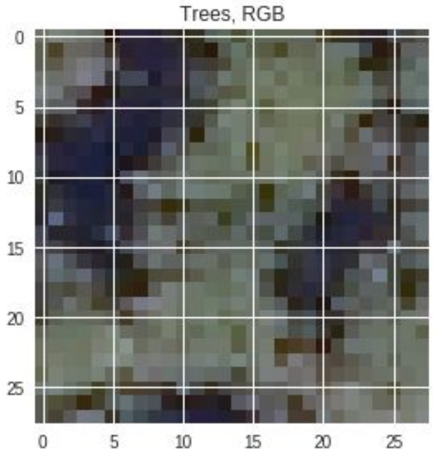
Barren, NIR



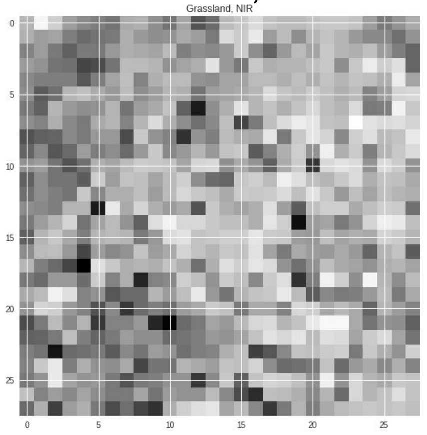
Barren, RGB



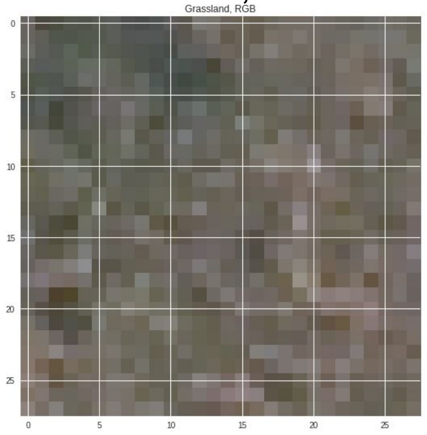


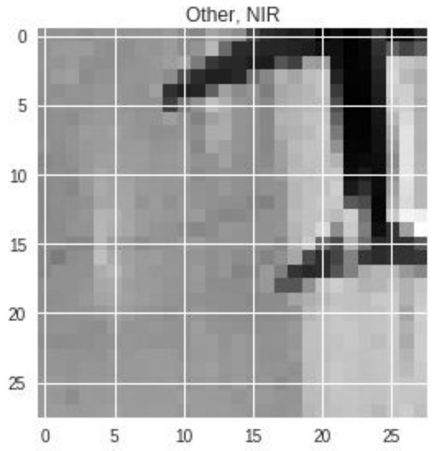


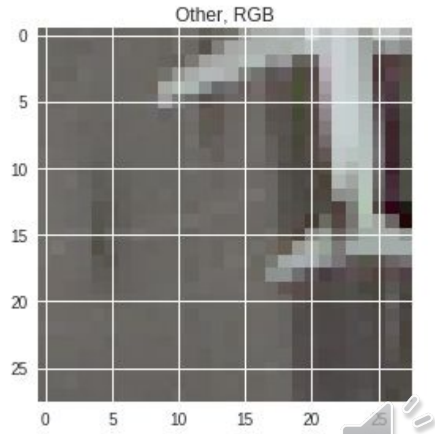
Grassland, NIR



Grassland, RGB







George

Images borrowed from dataset

Important Aspects Learned

- Memory management is important when working with large datasets
  - inefficiency of python’s memory management
  - reading data in chunks rather than loading the full dataset at once
- Start early, deep and complex neural networks take a very long time to train
- Optimizing to get the most of of the model is hard and takes time.

Gryffan Palmer

Alan Mai

Concluding Remarks

- VGG with transfer learning performs very well for classifying satellite images BUT, takes a long time to train :(
- Failed to replicate results from *Benchmarking Deep Learning Frameworks for the Classification of Very High Resolution Satellite Multispectral Data*
  - Late Fusion vs Early Fusion
  - Did not use full 224x224 image size of ImageNet
  - Training optimizations & increasing the number of epochs used while training
- Late Fusion with RGB + NIR has the potential to improve performance over just RGB.
  - Need to follow up with models (especially VGG trained on NIR) that have been trained longer or more data

Alan Mai

Follow ups

- correct image size to 224 x 224 to get the most out of pre trained ImageNet weights
- longer training with more epochs (> 200)
- early fusion of RGB + NIR
- more refined training techniques:
  - decreasing learning rate after number of epochs,
  - unfreezing convolutional layers after a number of epochs.
- Different weighting in Late Fusion:
  - Maybe have more weight emphasis on RGB
  - Weights between RGB and NIR trained while training.

Sources

1. https://csc.lsu.edu/~saikat/deepsat/
2. https://www.kaggle.com/crawford/deepsat-sat4
3. https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html
4. http://noiselab.ucsd.edu/ECE285/FinalProjects/Group16.pdf
5. http://cs231n.stanford.edu/reports/2017/pdfs/908.pdf
6. https://arxiv.org/pdf/1704.02965.pdf
7. http://inca38.nrsc.gov.in/Presentations_PDF/D2/86.pdf
8. https://arxiv.org/pdf/1704.02965.pdf
9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5017387/
10. https://arxiv.org/abs/1409.1556