

Land Classification and Building Detection

Said Mejia

Juan Pablo Montoya

Harold Gamarro

Joe Brian Malubay

Final Presentation
05/13/2021

Project overview

- Motivation
- Land Classification
 - Data Processing
 - Model Selection
 - Evaluation
 - Results
- Building Detection
 - Data Processing
 - Model Selection
 - Evaluation
 - Results

Land Classification Application

Automatic categorization of land cover is a key factor in forest preservation, extreme events damage assessment and urban growth.



Building Detection Application

Map features such as roads, building footprints, and points of interest are primarily created through manual techniques. Advancements in automated feature extraction can be applied to efforts such as humanitarian and disaster response, such as the events that took place with Hurricane Maria in Puerto Rico.

Before



After



Before



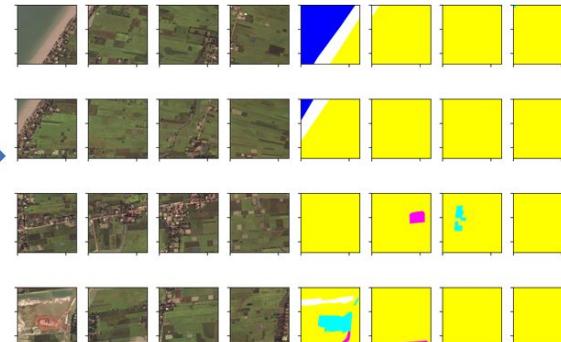
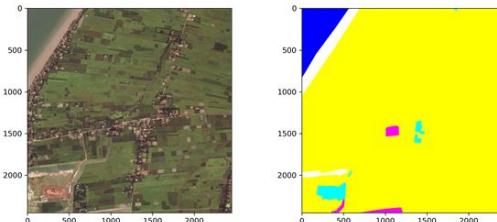
After



Describe the Data: landclass

- The DeepGlobe imagery is an image segmentation dataset collected by the DigitalGlobe's satellite.
- The images are 2448 wide x 2448 height and have a pixel resolution of 50cm.
- Each of the training images comes with an RGB mask representing the 7 different classes.
- Total of 803 images.

Preprocessing:

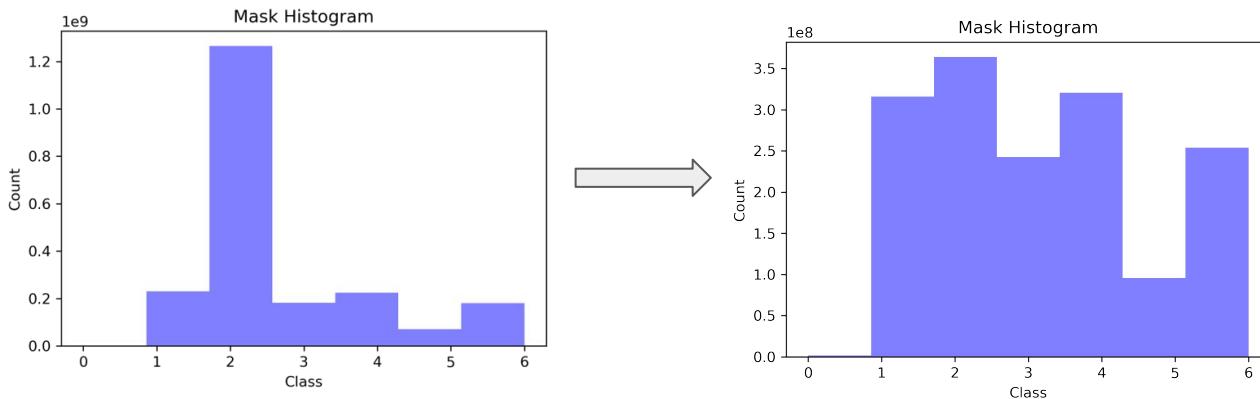


Dataset	Train	Val	Test	Total	Img Size
Raw	493	160	150	803	2448
Split	8221	2570	2065	12856	612

Balancing the Dataset: landclass

The agriculture land class is highly unbalanced in the DeepGlobe dataset.

- First method: Removing images with agriculture land class > 80% of the image.



Label	Class
0	unknown
1	urban_land
2	agriculture_land
3	rangeland
4	forest_land
5	water
6	barren_land

- Second method: Calculate the weight for each training example and incorporate them into the image generator.

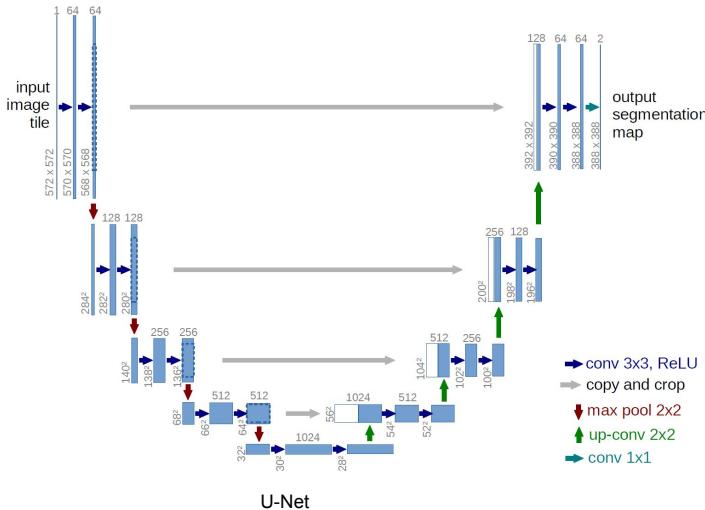
Models: LC

Encoders:

Pretrained Weights: ImageNet

Resnet50

VGG16



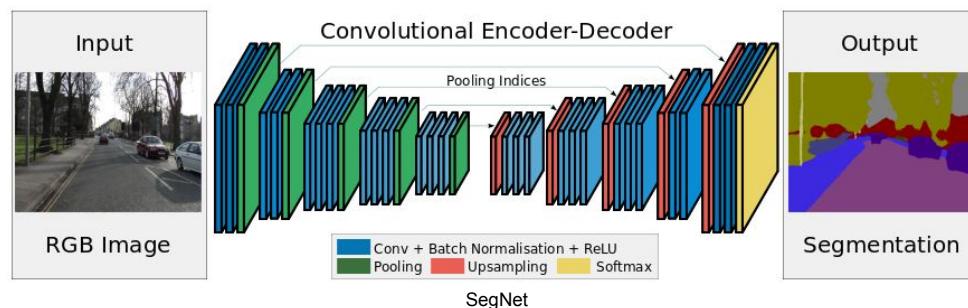
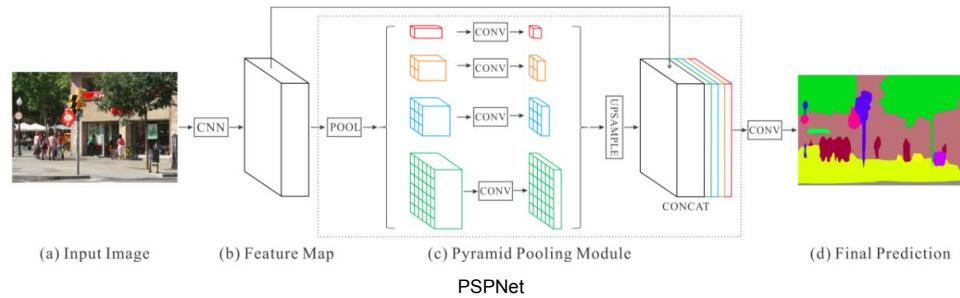
Decoders:

Unet

SegNet

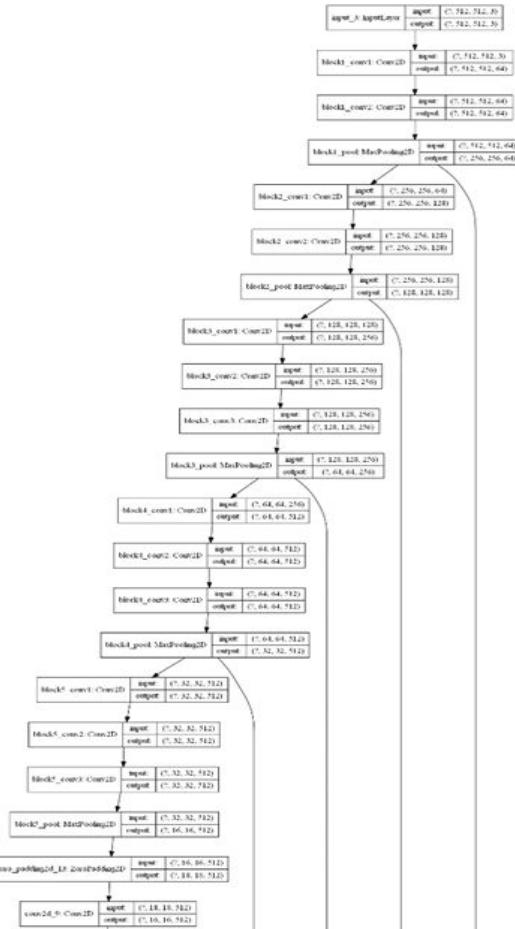
PSPNet

Models	Output
SegNet	512x512x7
Unet	512x512x7
PSPNet	144x144x7

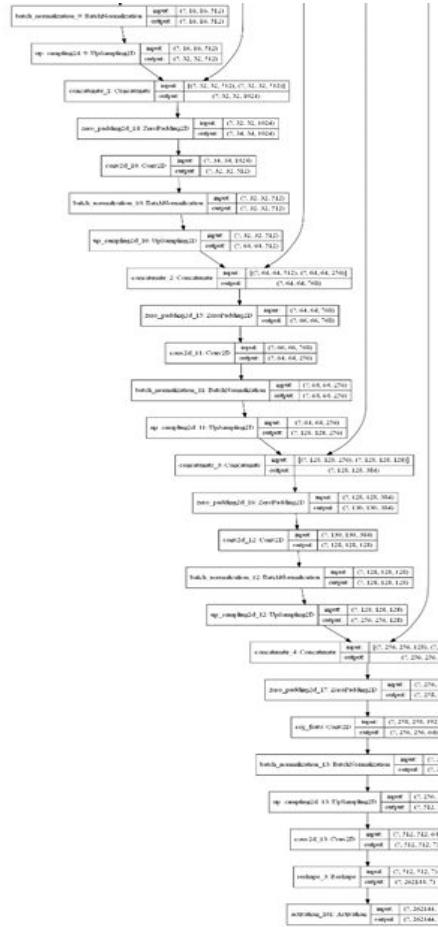


Encoder: VGG16

Encoder: VGG16



Decoder: U-Net



Callbacks and Metrics: LC

- ModelCheckpoint, EarlyStopping, ReduceROnPlateau
- Metrics: IOUScore and FScore
- Loss: dice_loss + 1*categoricalFocalLoss()

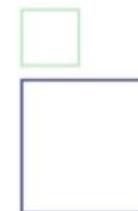


Area of Intersection

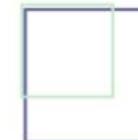
$$\text{IOU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$



Area of Union



IOU: 0
No Match



IOU: 0.4
Bad Annotation



IOU: 0.8
Good Annotation



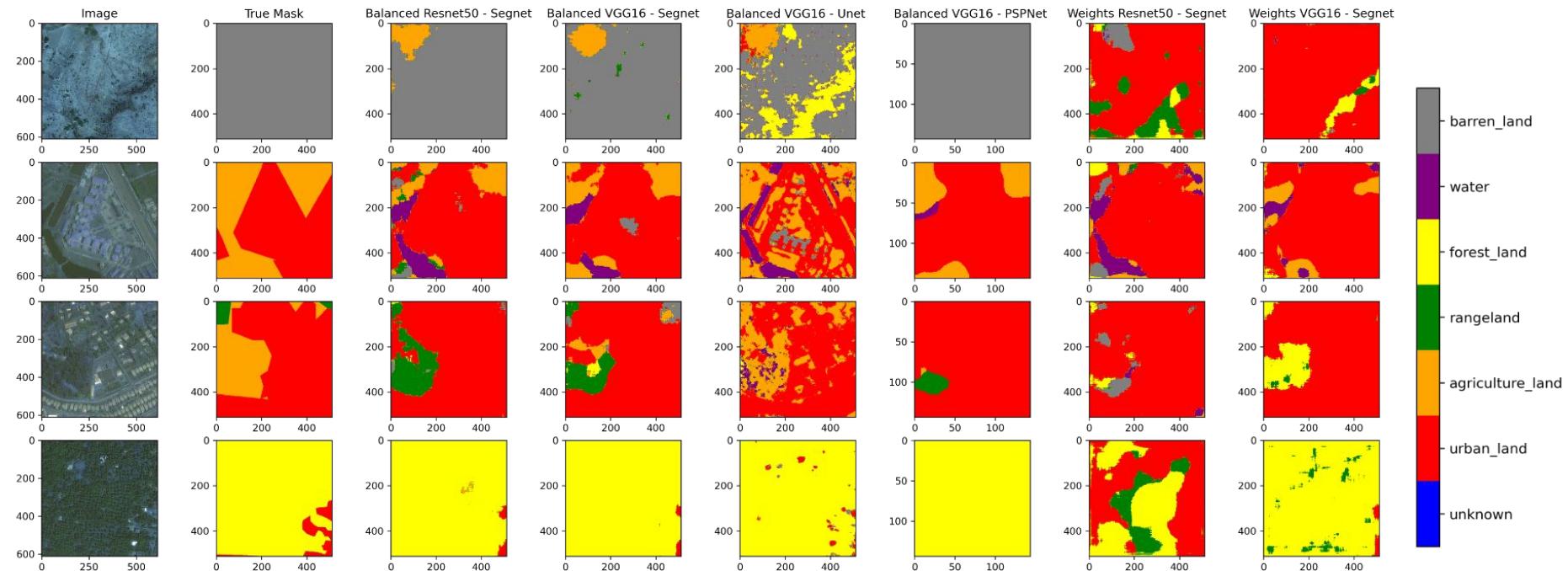
IOU: 0.9
Perfect Annotation

Transformation, Feature Selection, and Modeling: LC

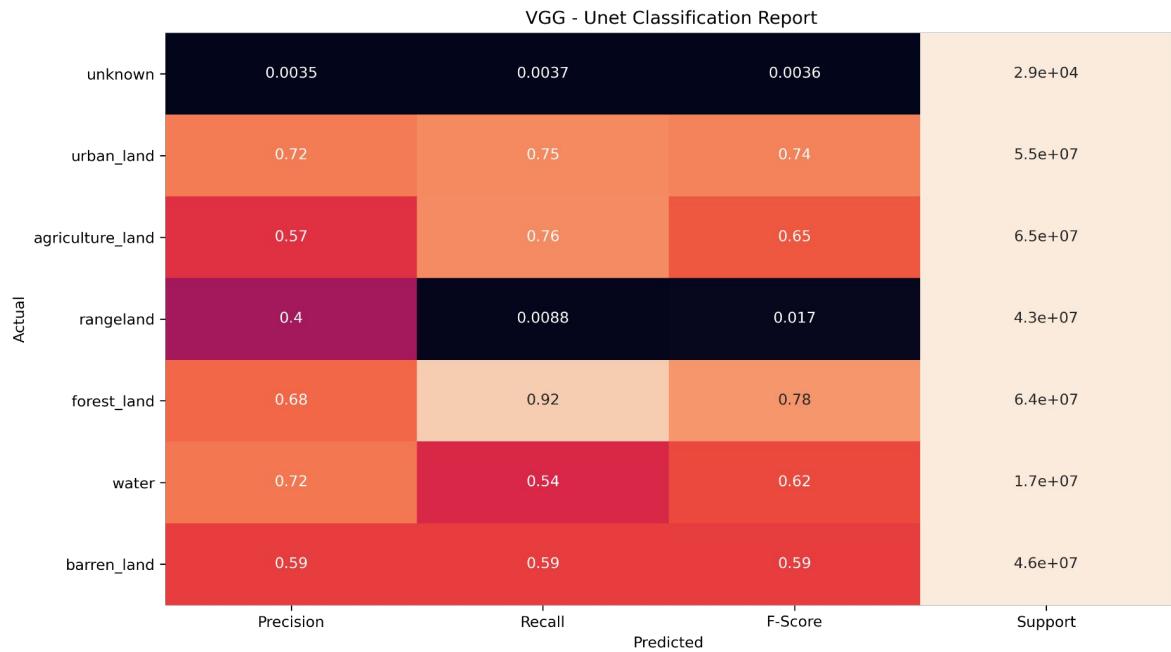
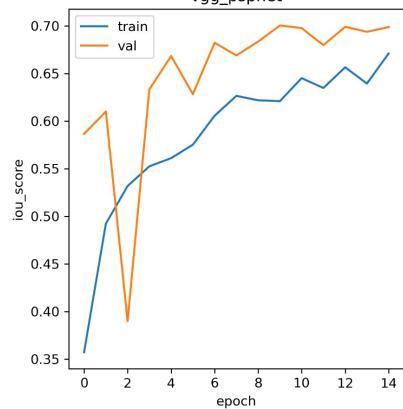
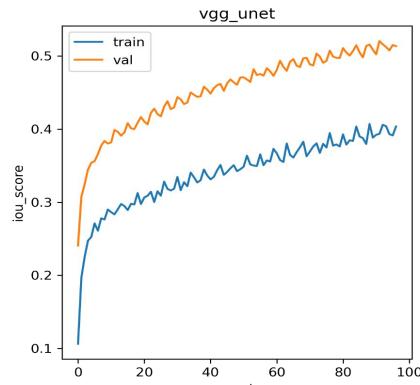
Models	Trainable Parameters	Non-trainable Parameters
VGG16-Unet	24,123,463	2,944
VGG16-Segnet	11,549,767	1,920
VGG16-PSPNet	17,111,367	5,120
Resnet50-Unet	43,117,831	56,064
Resnet50-Segnet	14,832,391	32,512
Resnet50-PSPNet	29,863,431	58,240

Models	IOU Score		F1 Score	
	Balanced	Weights	Balanced	Weights
VGG16-Unet	0.5204	0.57147	0.6625	0.6704
VGG16-Segnet	0.6534	0.68883	0.7770	0.7790
VGG16-PSPNet	0.70051	-	0.8086	-
Resnet50-Unet	0.5799	0.5845	0.7132	0.7203
Resnet50-Segnet	0.5949	0.64061	0.7292	0.7563
Resnet50-PSPNet	0.6348	-	0.7555	-

Example plots of the model



VGG - Unet Classification Report



Sentinel 2

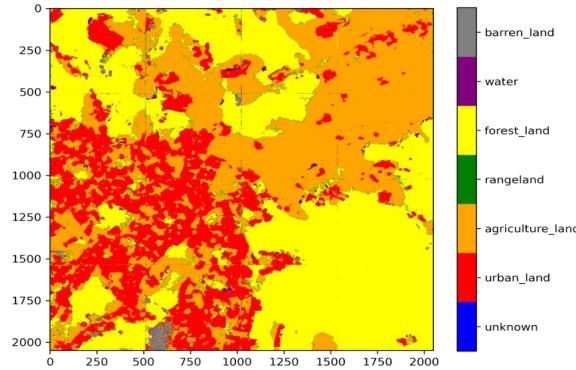
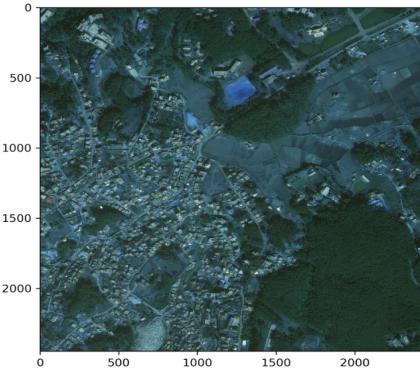


Aerial

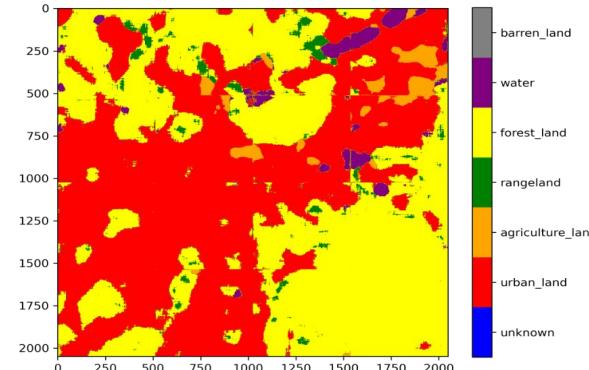
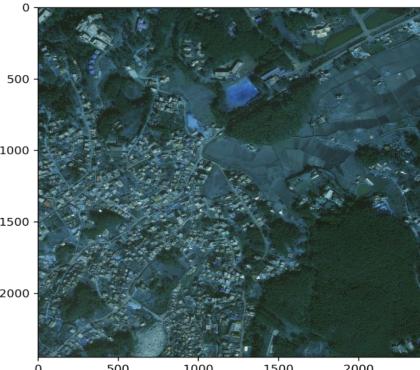


Predicting in a bigger area

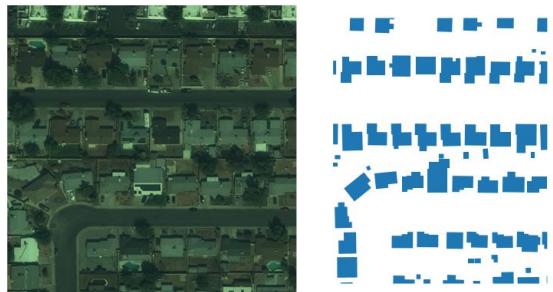
VGG - Unet
Balanced



VGG - Segnet
Weights



Building Detection: Dataset



Input Image

- 3 band geotiff
- 650 x 650 x 3
- ~ 3000 images

Output data

- Batches of 32 images
- Normalized data
- Rescaled to 512 x 512

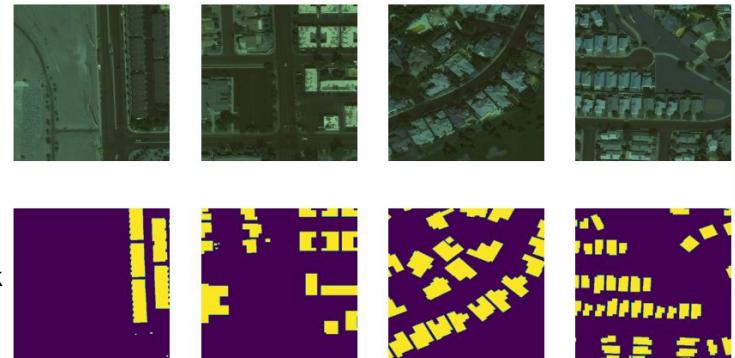
Progression

1. Local Laptop
2. Remote CPU server
3. Remote GPU server

Input mask

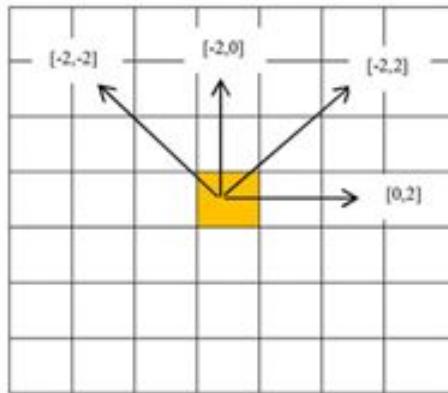
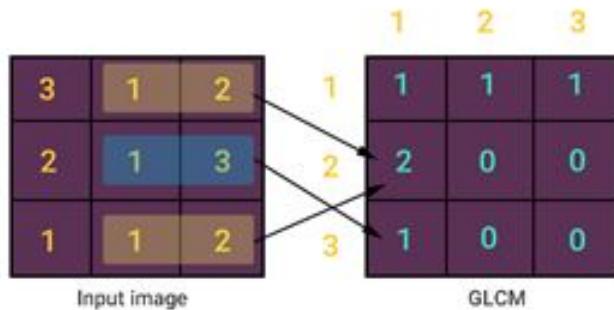
- Geojson
- 650 x 650 x 3

- 91 files of 32x512x512x3 for images
- 91 files of 32x512x512x1 for masks
- 75% Train, 15 % Validate, and 10 % Test



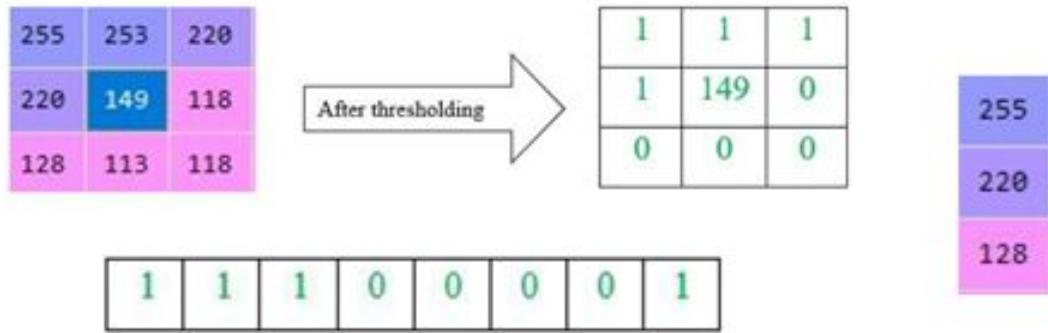
Random Forest

Haralick features

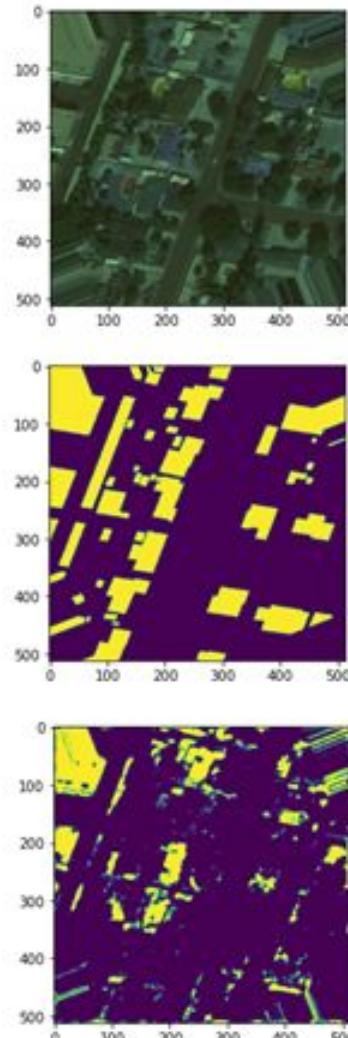


(a) $d = 2, \theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$

Local Binary Pattern



255	253	220
220	149	118
128	113	118



Building Detection Model: Encoder - Decoder CNN

Main changes from intro to Unet tutorial

Loss Function

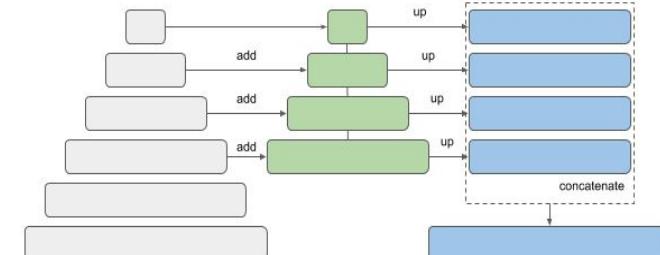
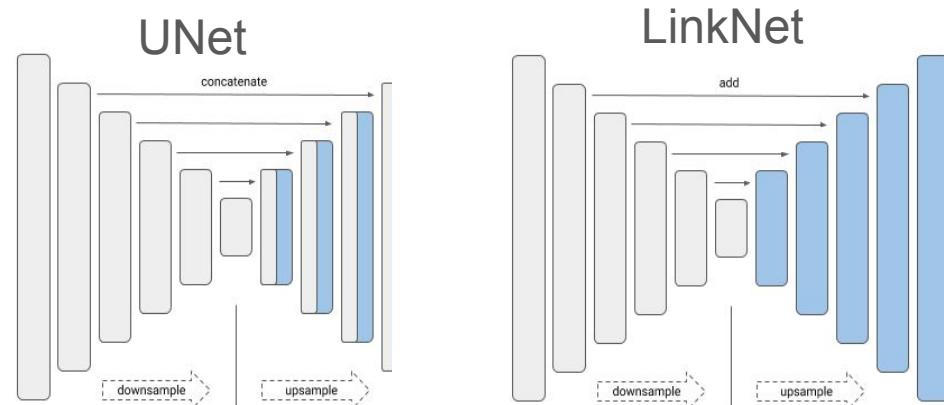
- Binary Focal Loss
- Dice Loss
- Total Loss = dice loss + (1 * focal loss)

Metrics

- IoU
- F1

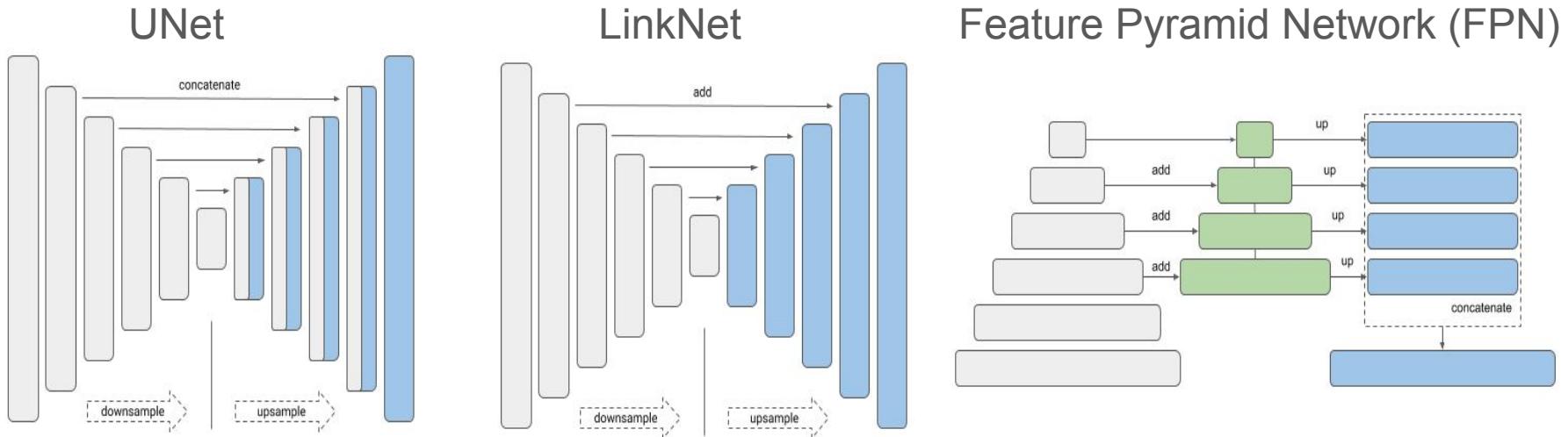
Modified Encoder & Pre-Weights

- Vgg16
- Efficientnet b3



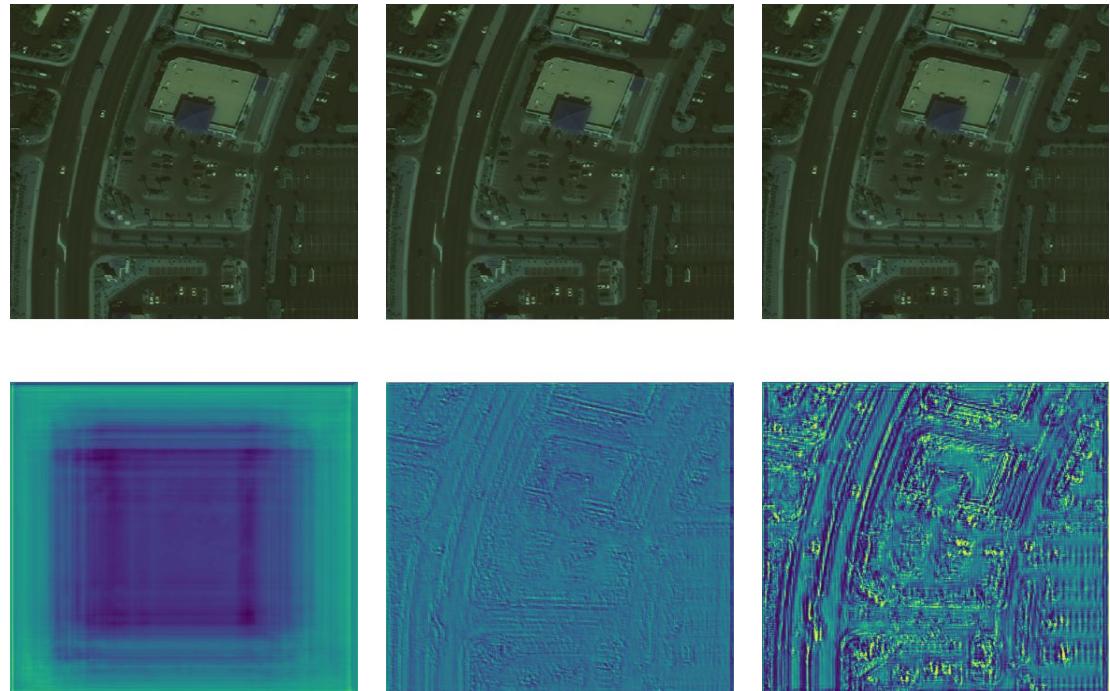
Feature Pyramid Network (FPN)

Building Detection Model: Encoder - Decoder CNN



Building Detection Model: Modified Encoder & Pre-Weights

- We explored the role of pre-weights
- Image shows the model outputs **before** any training
- VGG16 seems to outline edges
- Did not use Resnet50



Resnet50

Efficientnetb3

Vgg16

Building Detection Model: Pix2Pix

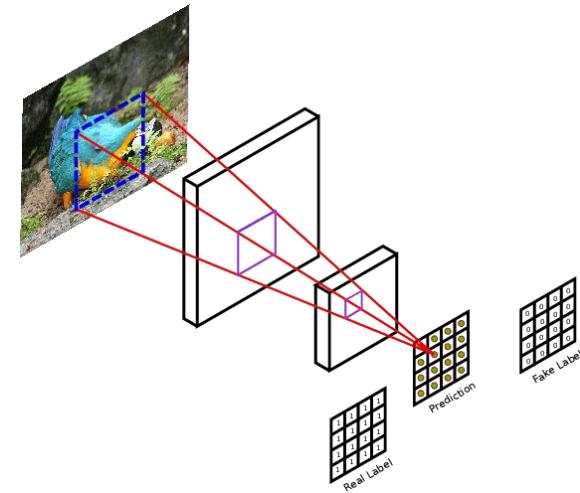
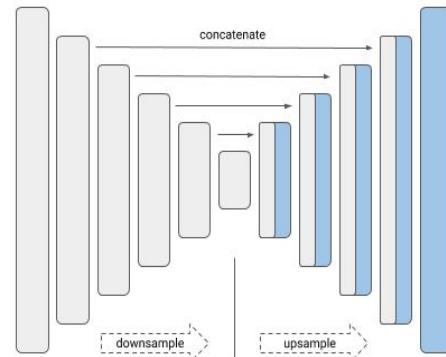
Conditional GAN to learn mapping from observed images

Generator

- Unet
- Sigmoid cross entropy loss of the generated images and an **array of ones**
- L1 loss between the generated image and the target image

Discriminator

- PatchGan
- Real loss is a sigmoid cross entropy loss of the **real images** and an **array of ones**
- Generated loss is a sigmoid cross entropy loss of the **generated images** and an **array of zeros**



Building Detection Model: Pix2Pix

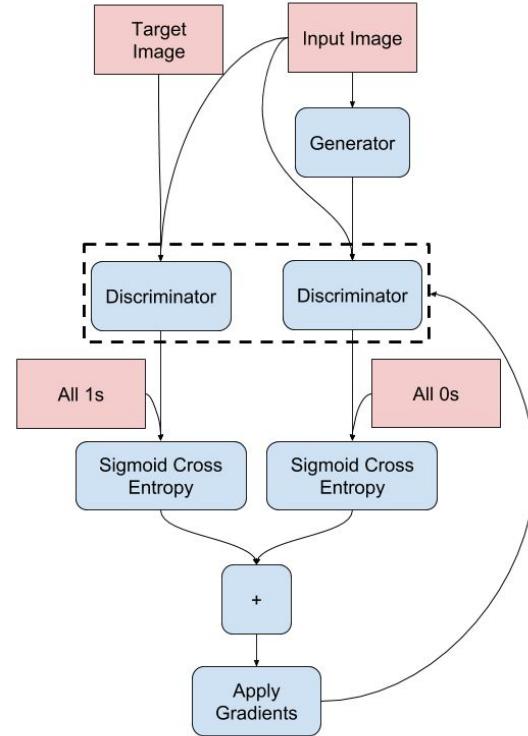
Conditional GAN to learn mapping from observed images

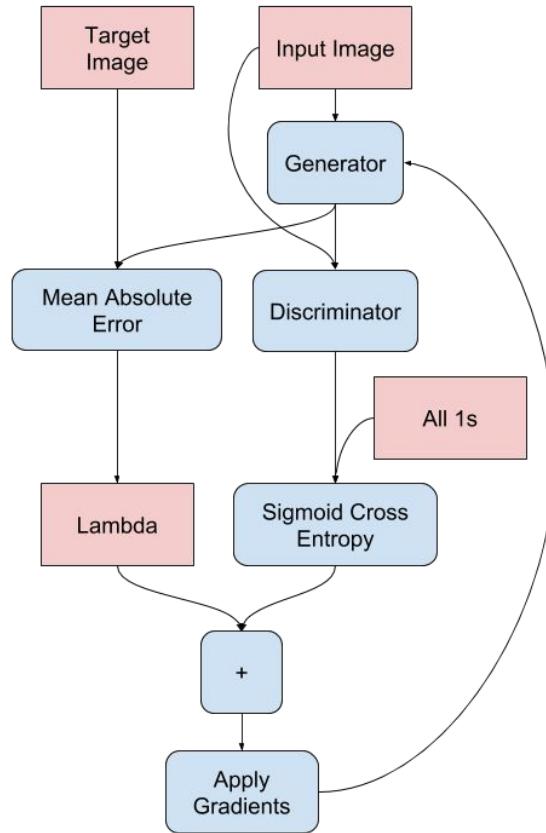
Generator

- Unet
- Sigmoid cross entropy loss of the generated images and an **array of ones**
- L1 loss between the generated image and the target image

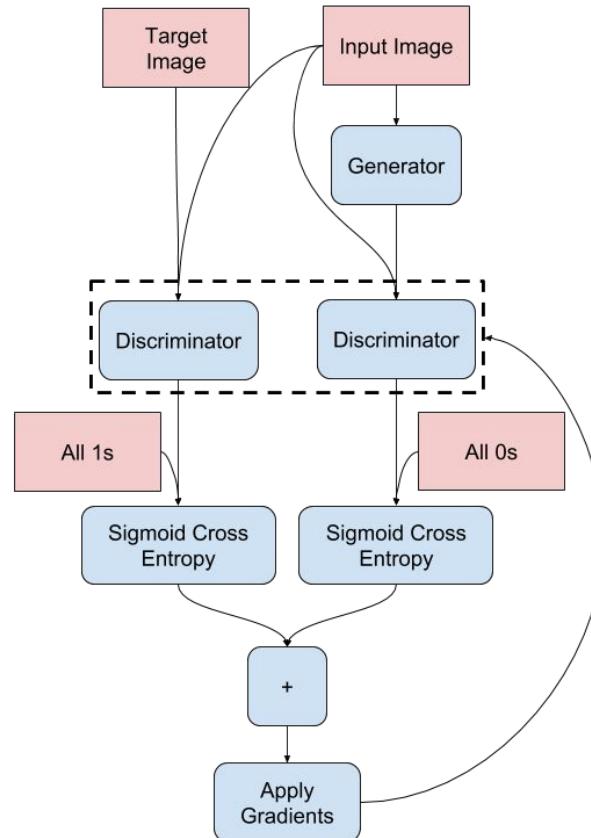
Discriminator

- PatchGan
- Real loss is a sigmoid cross entropy loss of the **real images** and an **array of ones**
- Generated loss is a sigmoid cross entropy loss of the **generated images** and an **array of zeros**





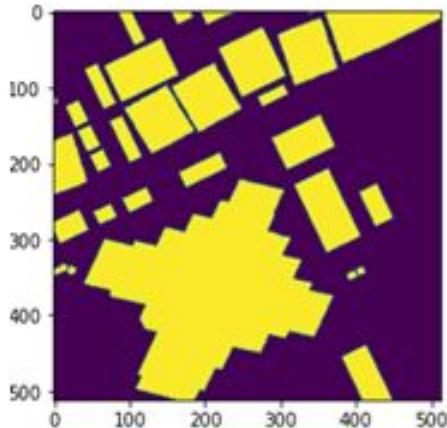
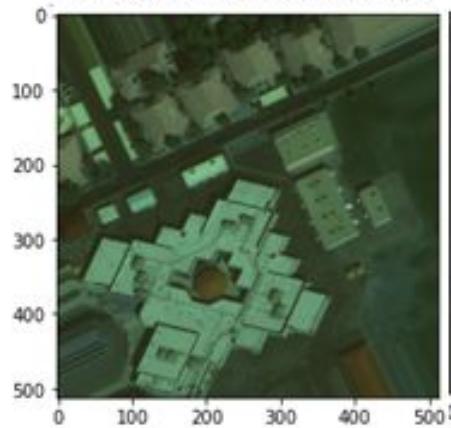
A- Generator Loss



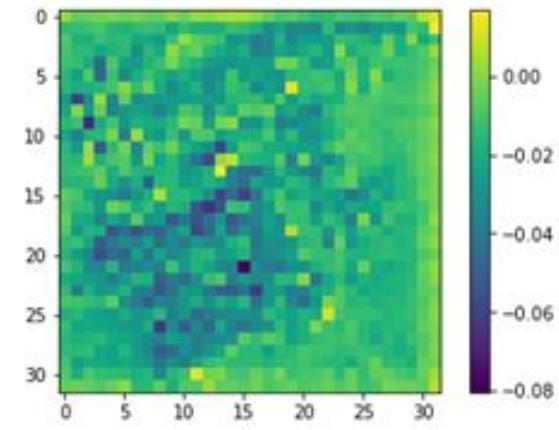
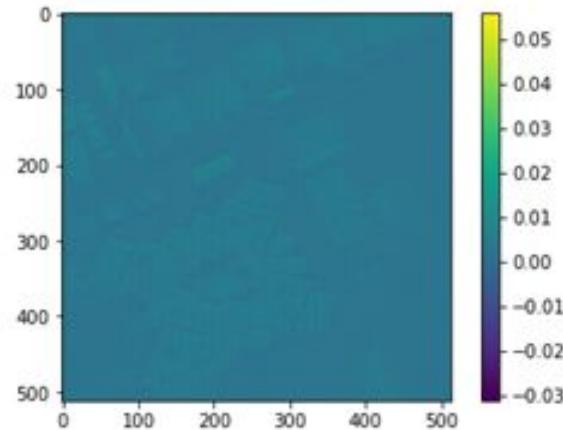
B- Discriminator Loss

Pix2Pix 0th Model

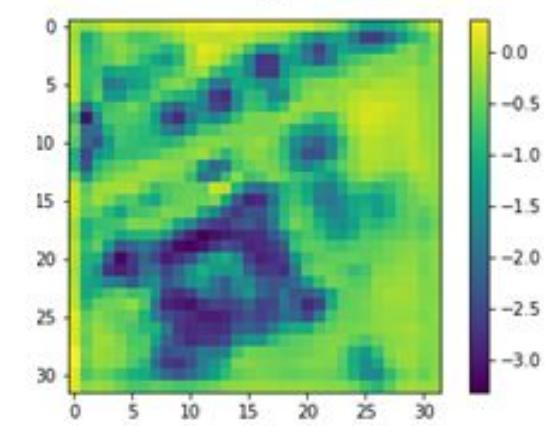
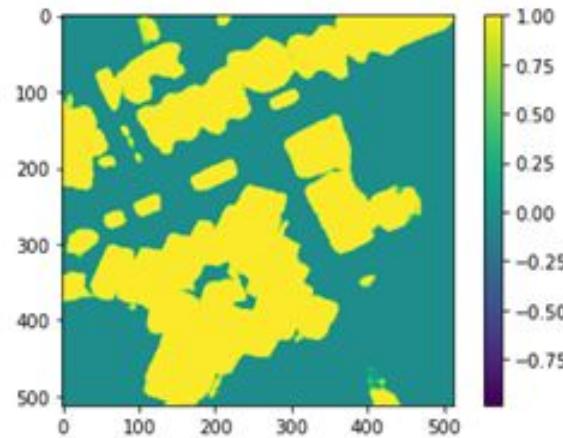
Image and Ground Truth



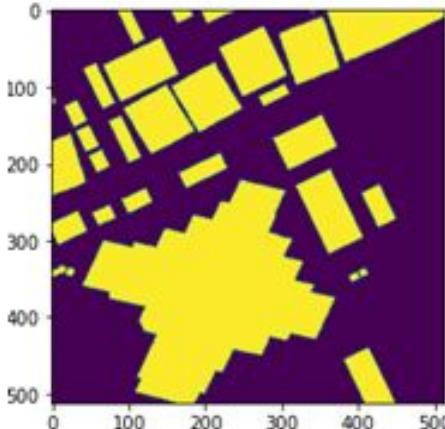
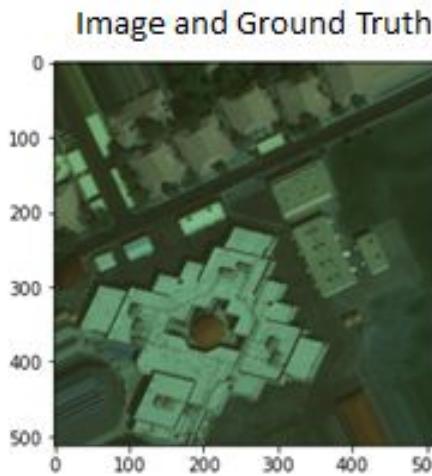
Generator and Discriminator before training



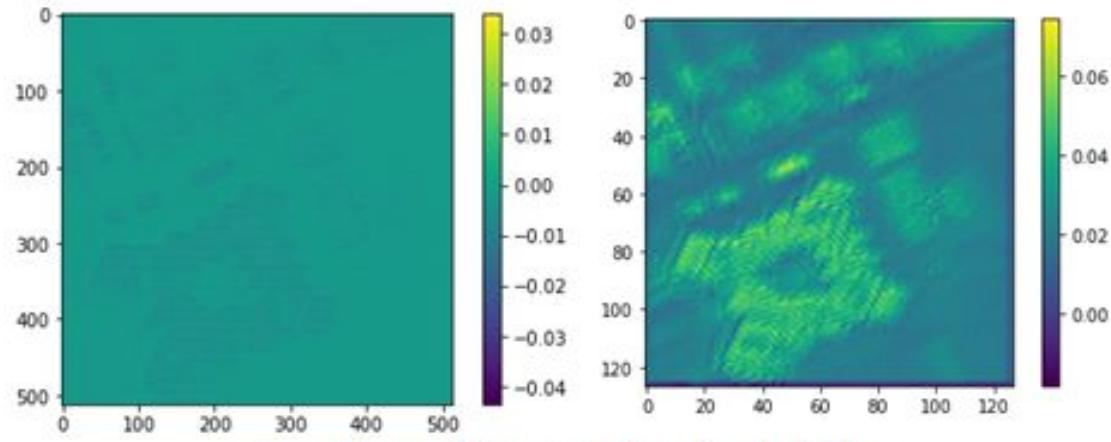
Generator and Discriminator after training



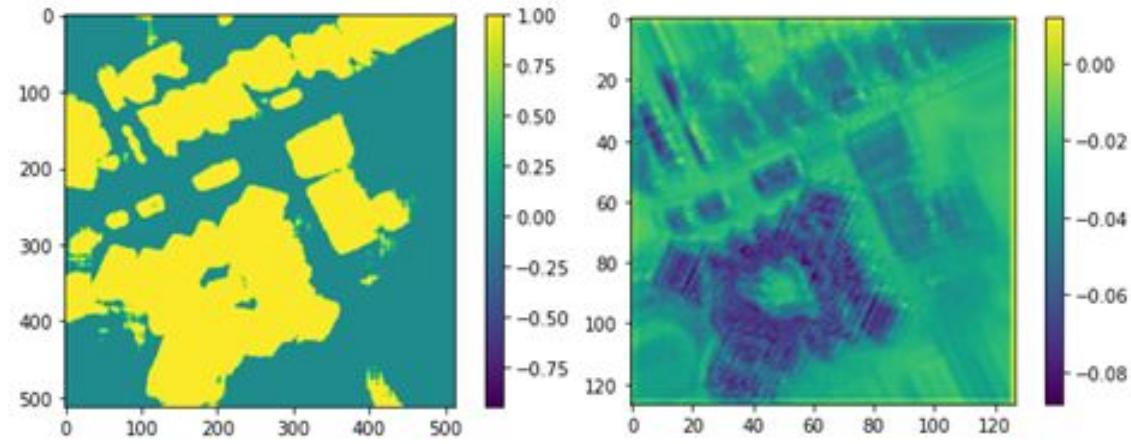
Pix2Pix 0' Model



Generator and Discriminator before training

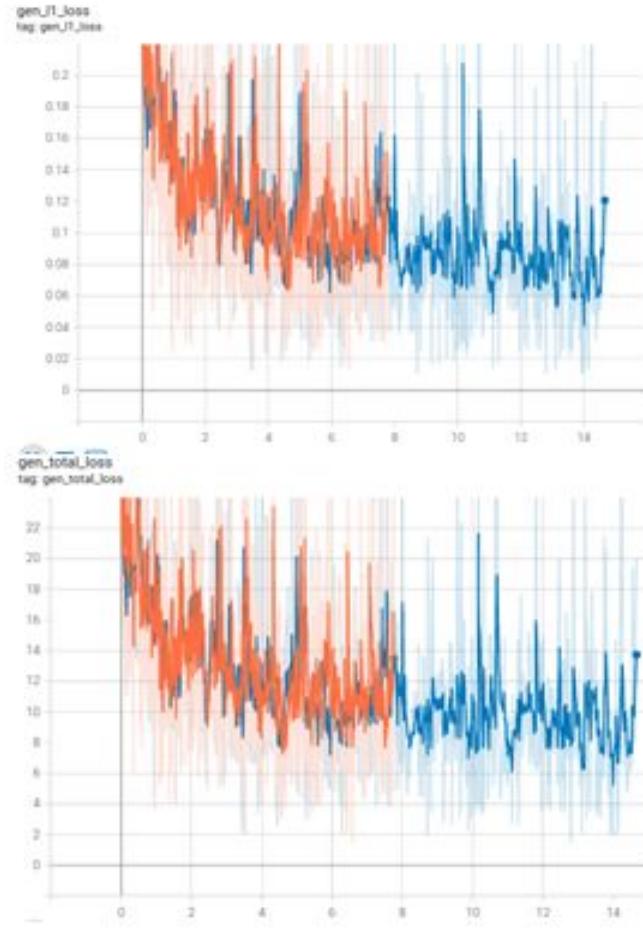
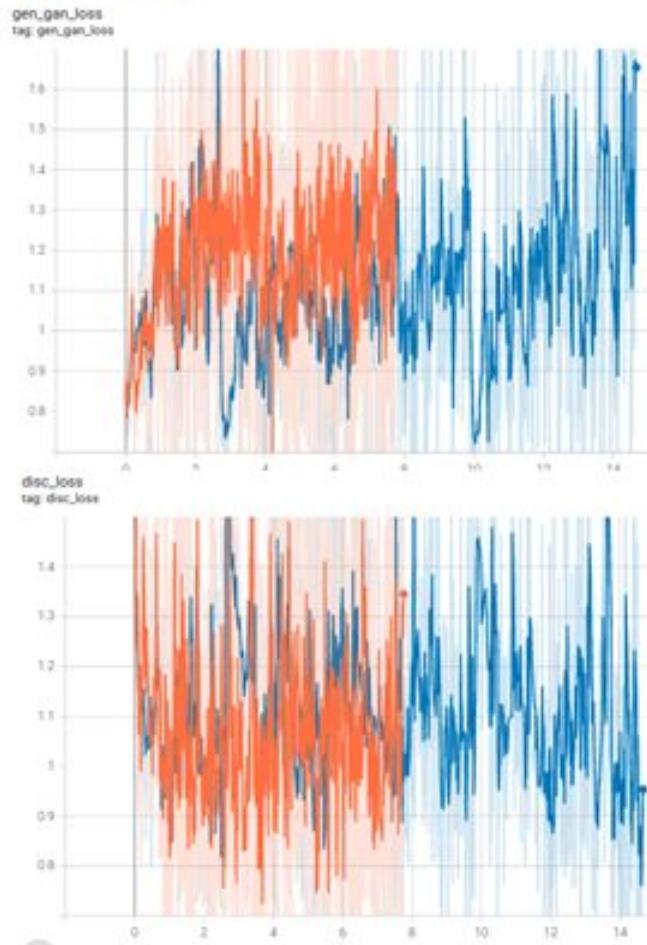


Generator and Discriminator after training



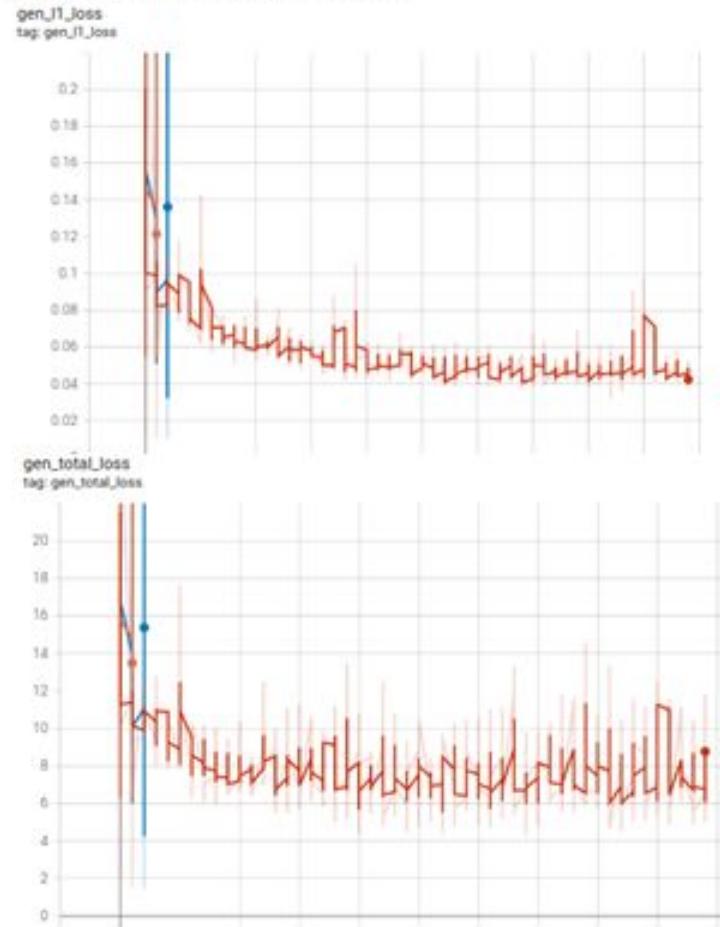
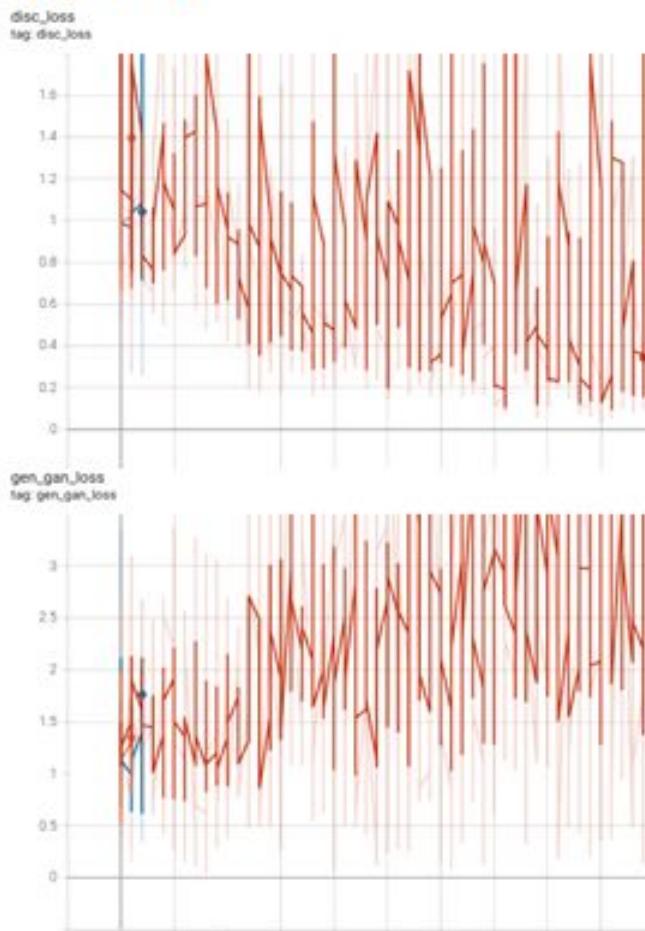
Pix2Pix

0th model: orange, 0'model: blue



Pix2Pix

0th model: orange, 0' model: blue, V1 model: red



Building Detection Model: Pix2Pix Model Versions

Generator

- Unet
 - V0 , V0' and V1 - Simple Unet no pre-weights and tanh activation
 - V2 - Vgg16 Unet with Sigmoid activation

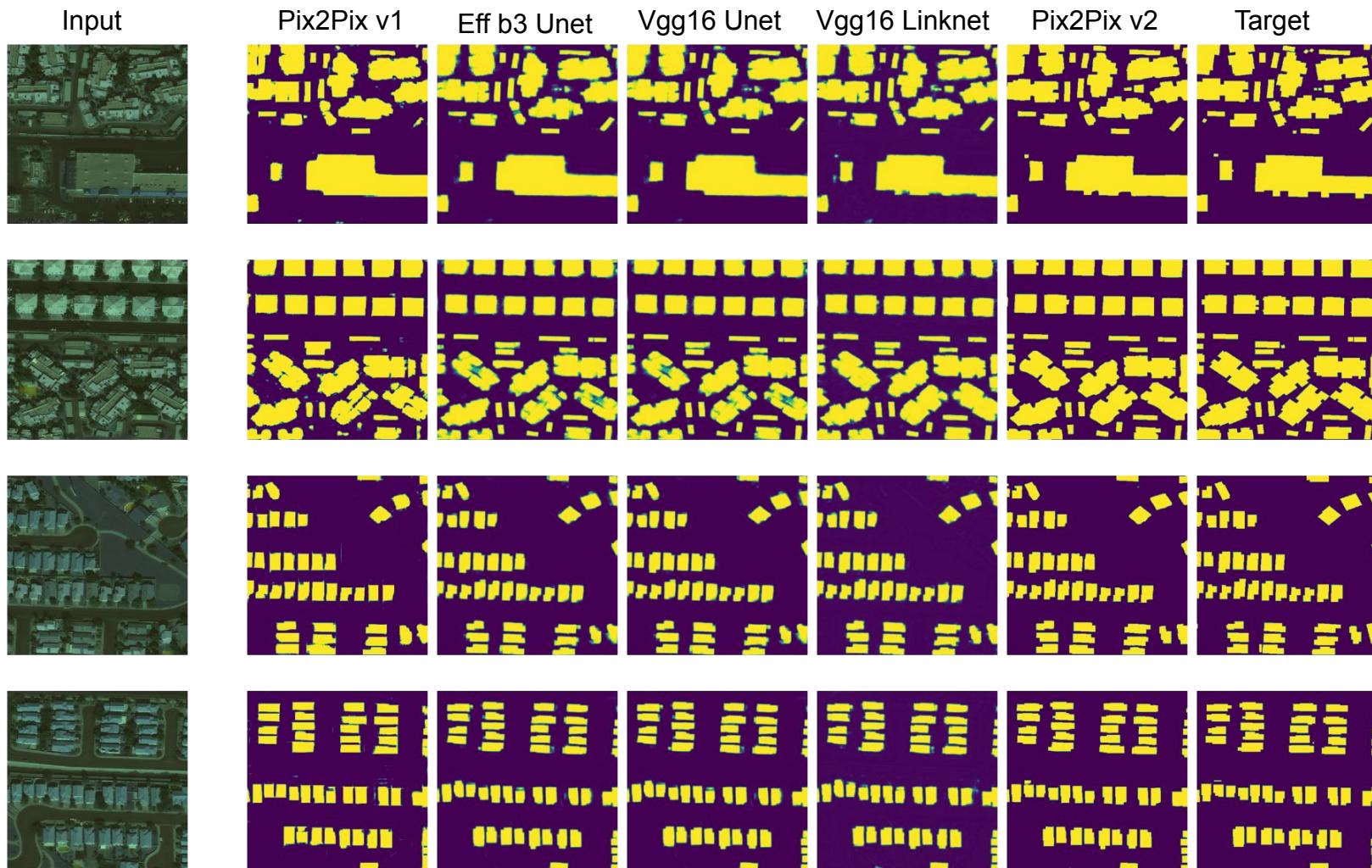
Discriminator

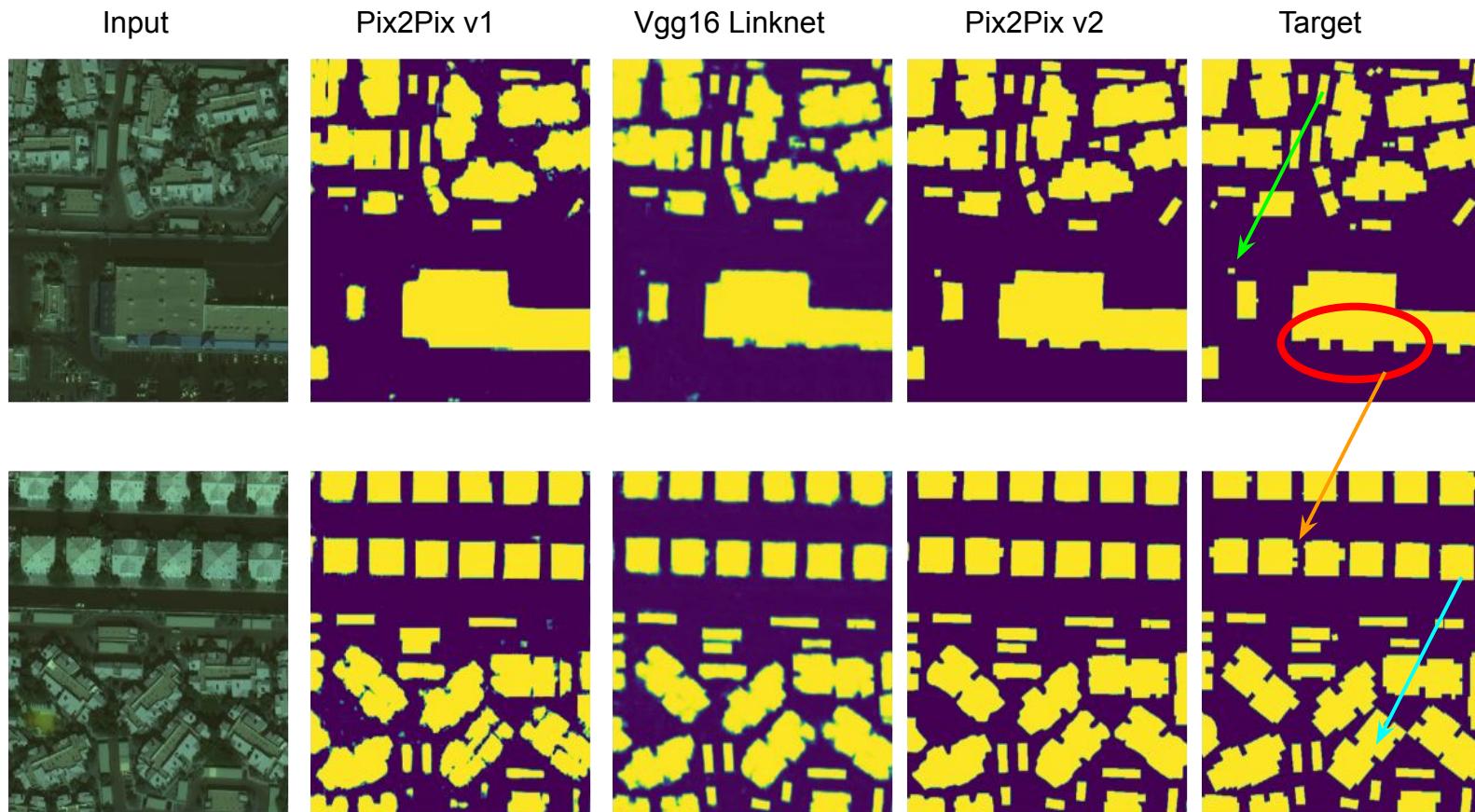
- PatchGan
- Each patch of the output classifies a specific portion of the input image
 - V0 - 32x32 patch classifies 110x110 of input
 - V0' - 128x128 patch classifies 40x40 of input
 - V1 - 30x30 patch classifies 120x120 of input
 - V2 - 60x60 patch classifies 70x70 of input

Building Detection: Model evaluation

```
[RESULTS] IOU_score: 0.78
[RESULTS] Accuracy: 0.88
[RESULTS] Precision: 0.87
[RESULTS] Recall: 0.88
[RESULTS] F1: 0.87
-----
Processing time: 43675.29942846298
```

Model version	mIOU	Recall	Precision	F1
Random Forest	-	88.00	87.00	87.49
pix2pix v1	87.62	89.07	89.39	89.23
efficientnetb3 - Unet	88.96	93.33	87.69	90.42
VGG16 - FPN	89.44	91.99	89.87	90.91
VGG16 - Unet	89.66	92.25	90.02	91.12
VGG16 - Linknet	89.71	91.61	90.76	91.18
pix2pix v2	94.26	95.04	95.44	95.24





Code Collaboration

- Common Data loader for each task
- Same loss functions
- Same model metrics
- Running all models on GPU supercomputers
- Group troubleshooting between all models

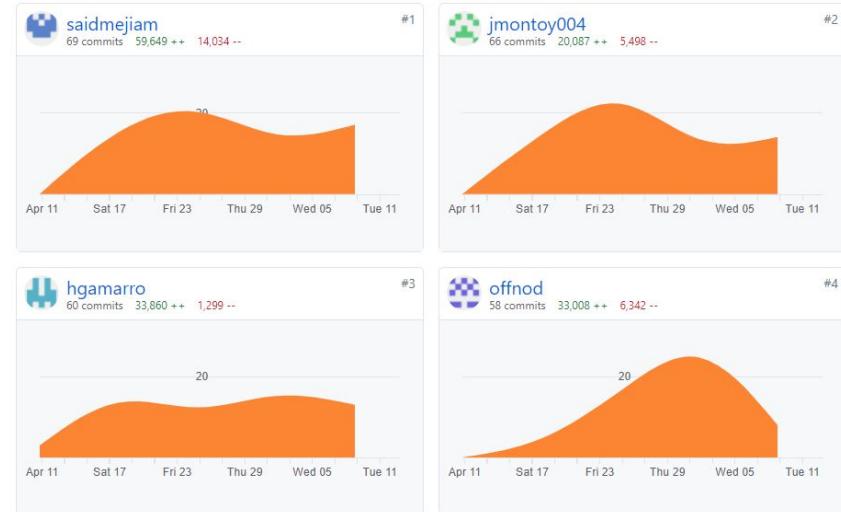
Notebook total

HG: 20

JB: 14

JM: 15

SM: 25



total commits - 293

Thanks

Questions?