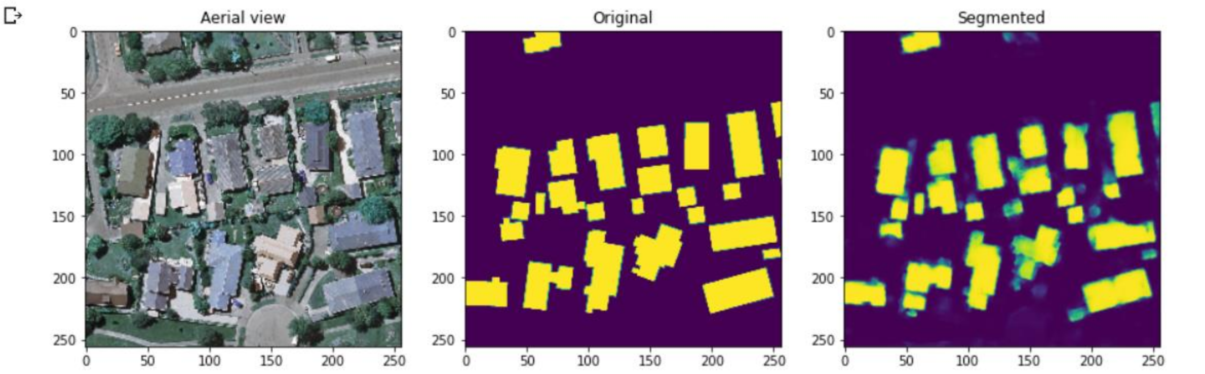
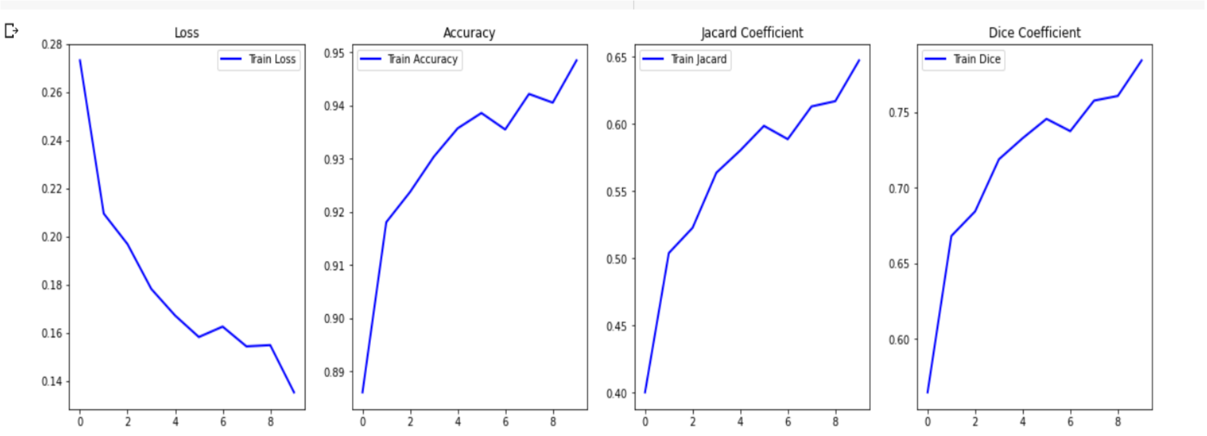
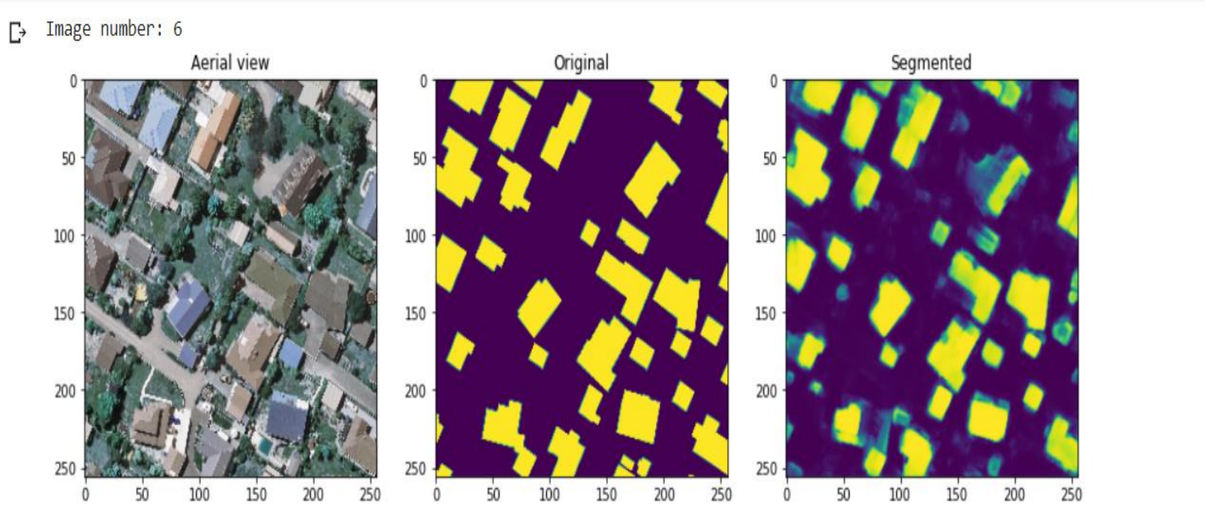
MODEL META DATA

Model Name: u-net-1000images-baseline
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 10
Batch size: 8
Learning rate: 0.01
No of images: 1008
Performance Metrics: IOU, Accuracy
Colab or Kaggle: Colab
Filename: Basic U-Net
Results: IOU - 18.06, Accuracy - 95.81
Comments: Scaling done by directly dividing by 255



Model Name: unet-youtube-1000images-10epochs
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 10
Batch size: 16
Learning rate: 0.01
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, Accuracy
Colab or Kaggle: Colab
Filename: Youtube U-Net
Results: IOU - 64.73, Dice Coefficient - 78.43 Accuracy - 94.84
Comments: Used MinMaxScaler, Referred Digital Sreeni video

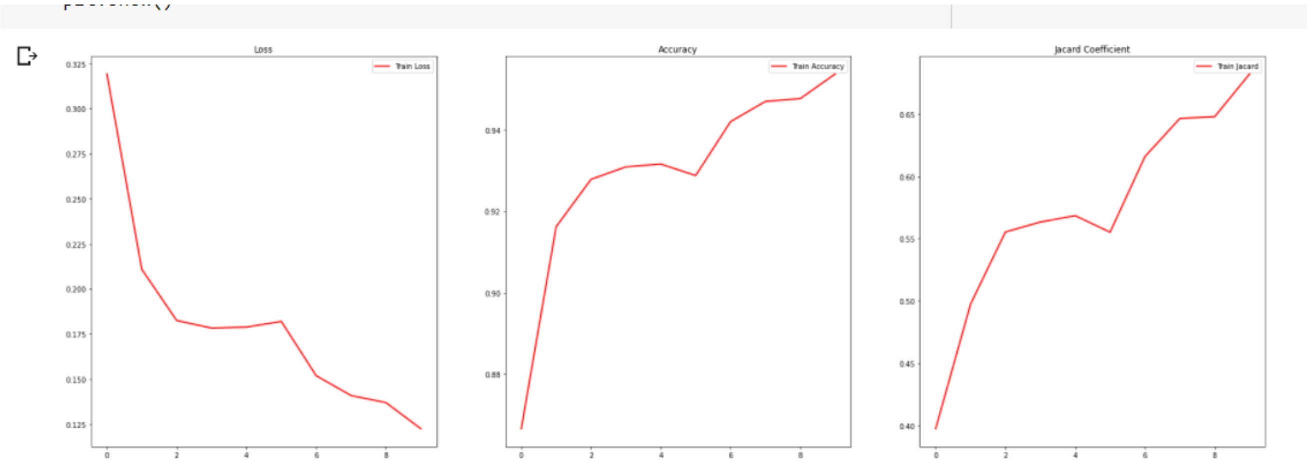


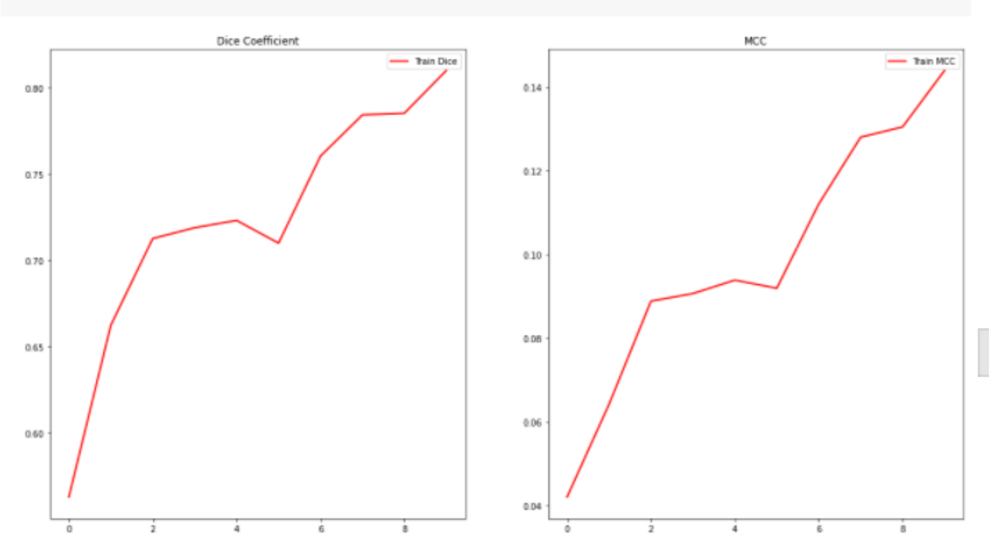


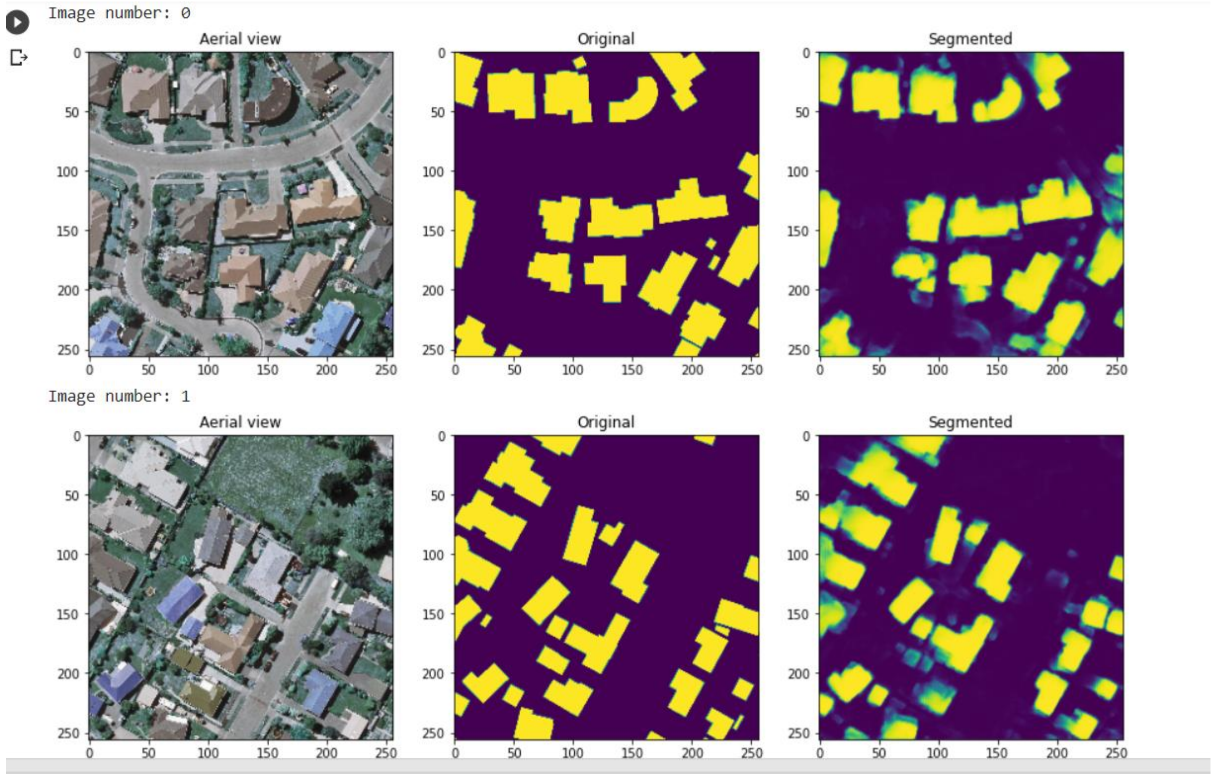
```
Image number: 6
```

```
[ ]  # tf.cast(inp, tf.float32)
```

Model Name: unetmcc-youtube-1000images-10epochs
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 10
Batch size: 16
Learning rate: 0.01
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Colab
Filename: Youtube U-Net
Results: IOU - 68.22, Dice Coefficient - 80.99, MCC - 14.40, Accuracy - 95.37
Comments: Used MinMaxScaler, Added MCC also as performance metrics, Referred Digital Sreeni video

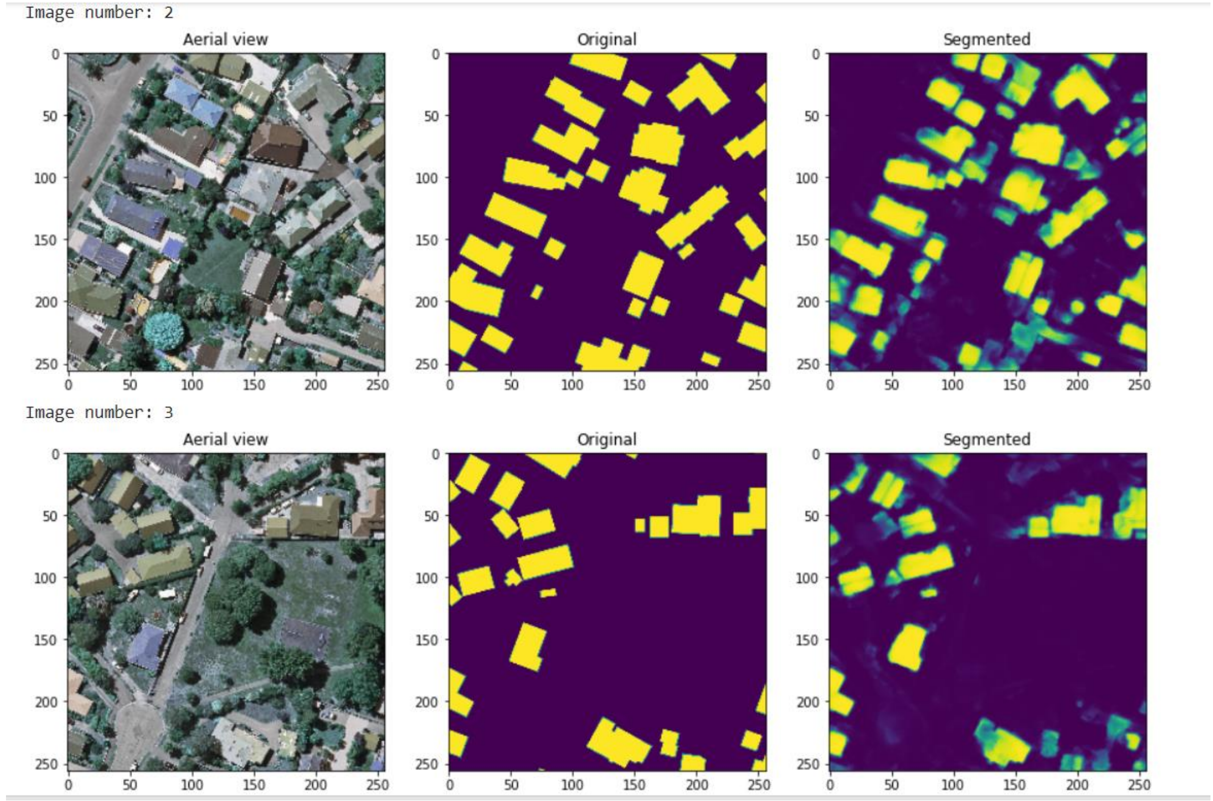






```
Image number: 0
```

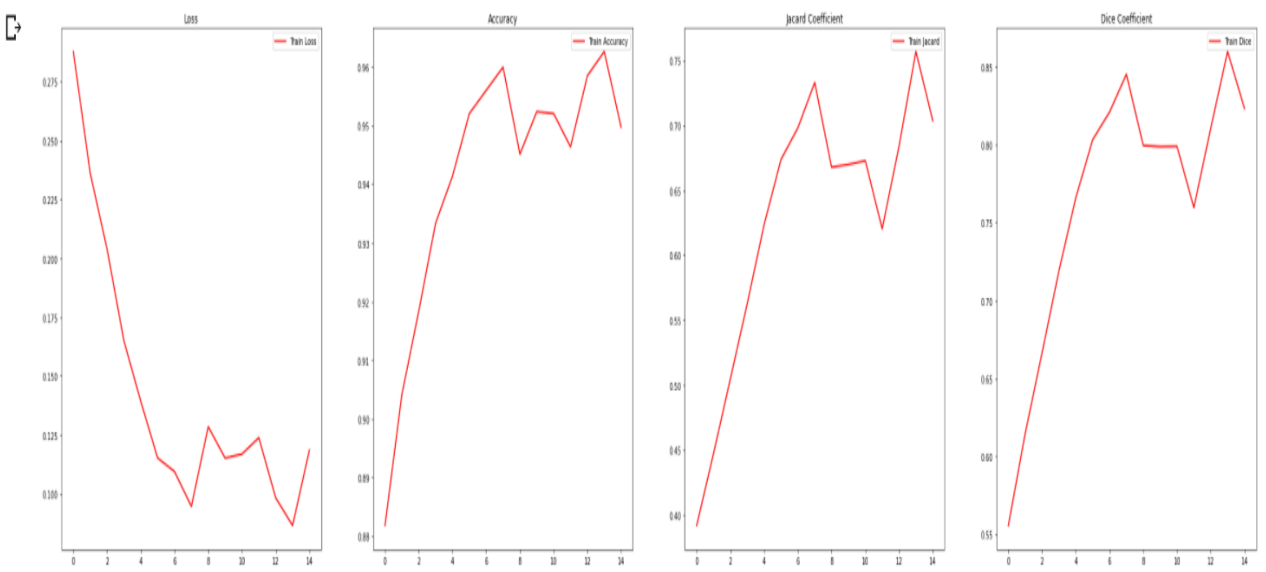
```
Image number: 1
```



```
Image number: 2
```

```
Image number: 3
```

Model Name: resumable-model
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 15
Batch size: 8
Learning rate: 0.001
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Colab
Filename: Youtube U-Net
Results: IOU - 70.36, Dice Coefficient - 82.34, MCC - NaN, Accuracy - 94.98
Comments: Same as above, Tried How to start, stop resume training of a model using keras-buoy

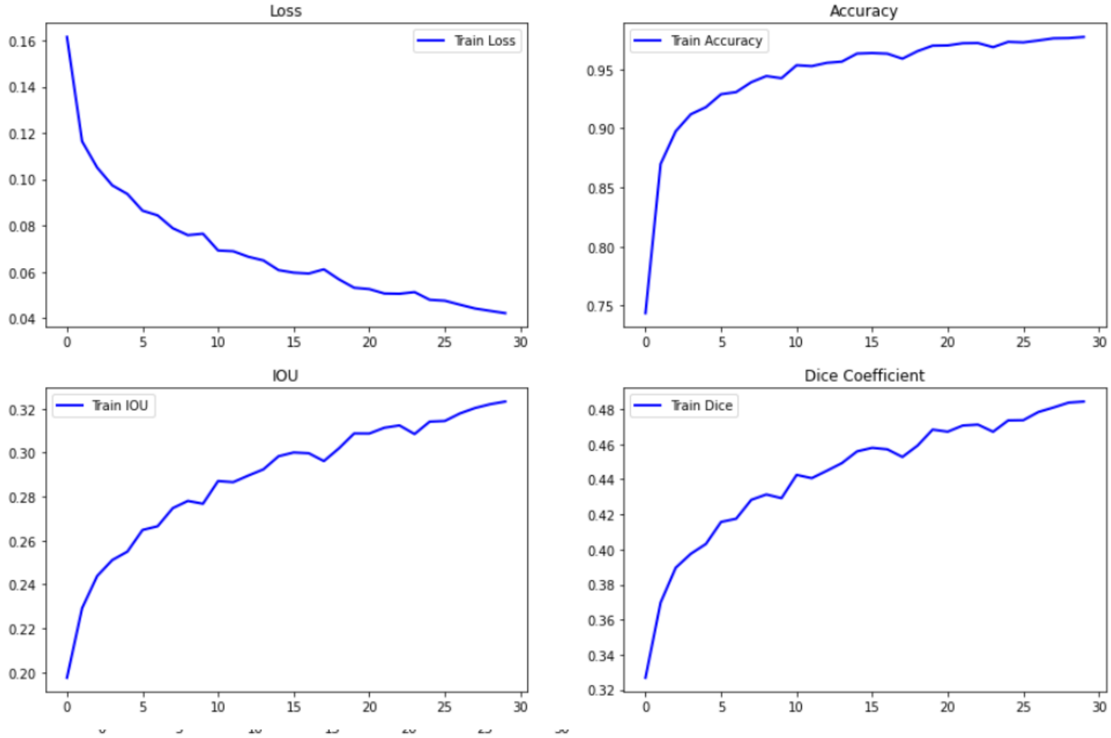


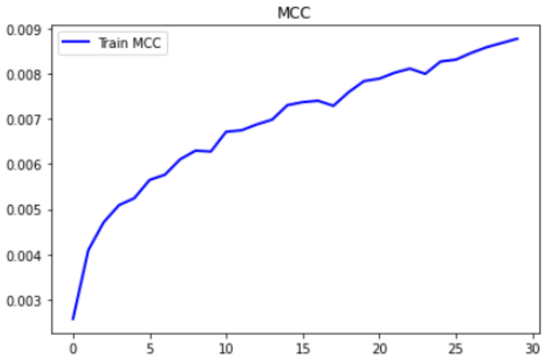
Model Name: 1500images-30epochs
Architecture: MultiRes U-Net
Numpy or Image Generator: Numpy
Epochs: 30
Batch size: 8
Learning rate: 0.0001
No of images: 1548
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Kaggle
Filename: 1500 Numpy MultiRes Version 1 - Numpy MultiRes, U-Net
Results: IOU - 30.21, Dice Coefficient - 45.87, MCC - 0.89, Accuracy - 97.53, Loss - 0.3288
Comments: Tried MultiRes U-Net with numpy array.

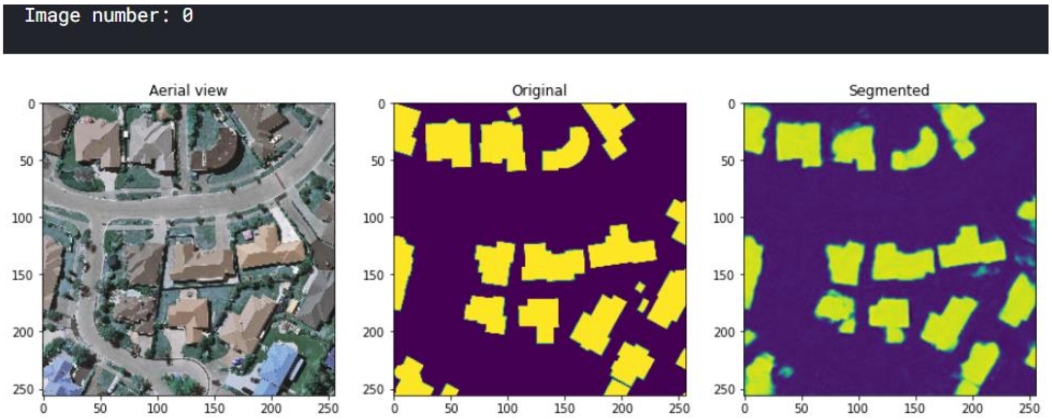
....

Model Name: 1500images-npy-100epochs
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 100
Batch size: 8
Learning rate: 0.0001
No of images: 1548
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Kaggle
Filename: 1500 Numpy MultiRes Version 1 - Numpy MultiRes, U-Net
Results: IOU - 86.81, Dice Coefficient - 92.83, MCC - 48.97, Accuracy - 98.28, Loss - 0.0339
Comments: U-Net with numpy array. Ran faster than in Colab. Took 48s for each epoch in Kaggle compared to 2mins in Colab.

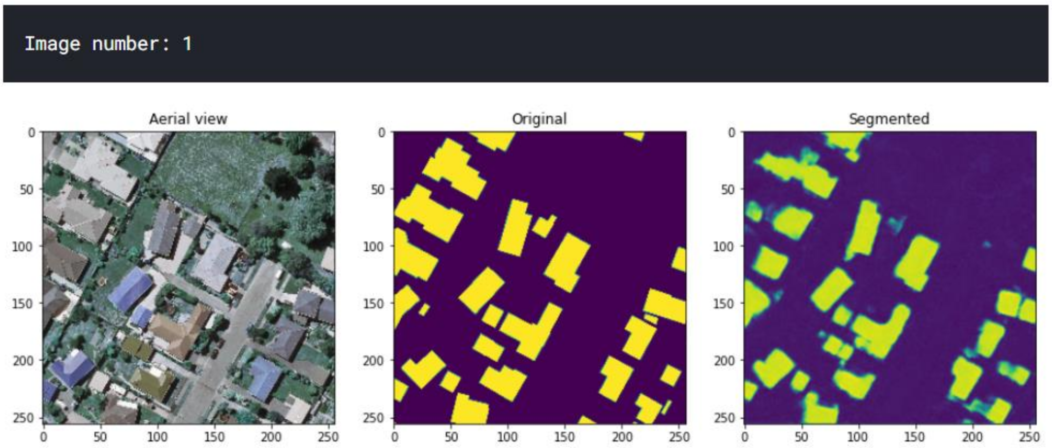
Model Name: focalloss-30epochs
Architecture: MultiRes U-Net
Numpy or Image Generator: Numpy
Epochs: 100
Batch size: 8
Learning rate: 0.0001
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Kaggle
Filename: Keras Data Generator Version 1 - multires-30
Results: IOU - 32.33, Dice Coefficient - 48.43, MCC - 0.88, Accuracy - 97.73, Loss - 0.0422
Comments: MultiRes with binary focal loss instead of binary cross entropy loss.



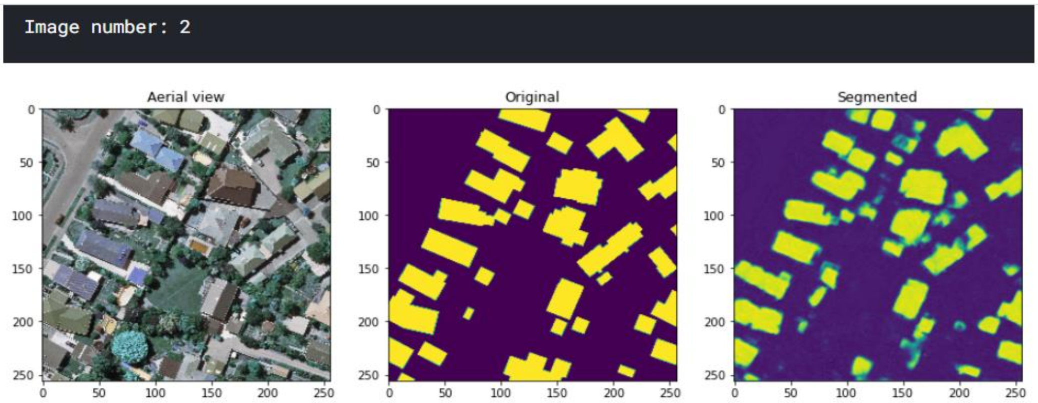




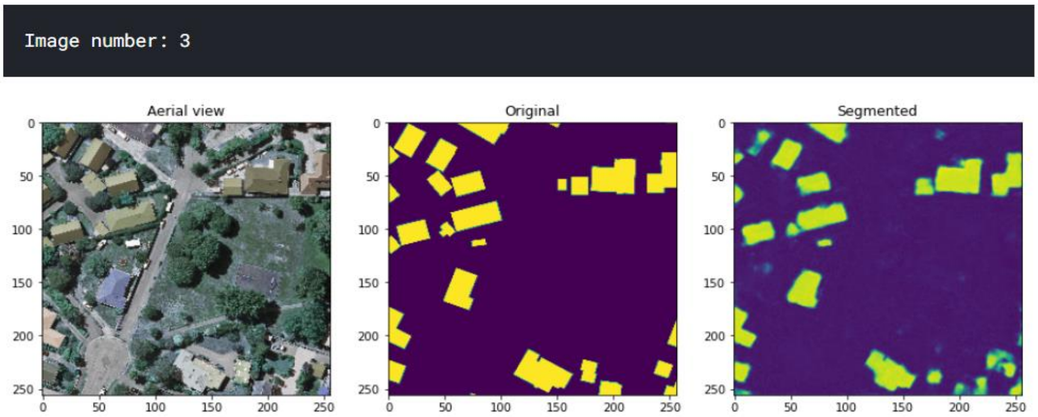
```
Image number: 0
```



```
Image number: 1
```

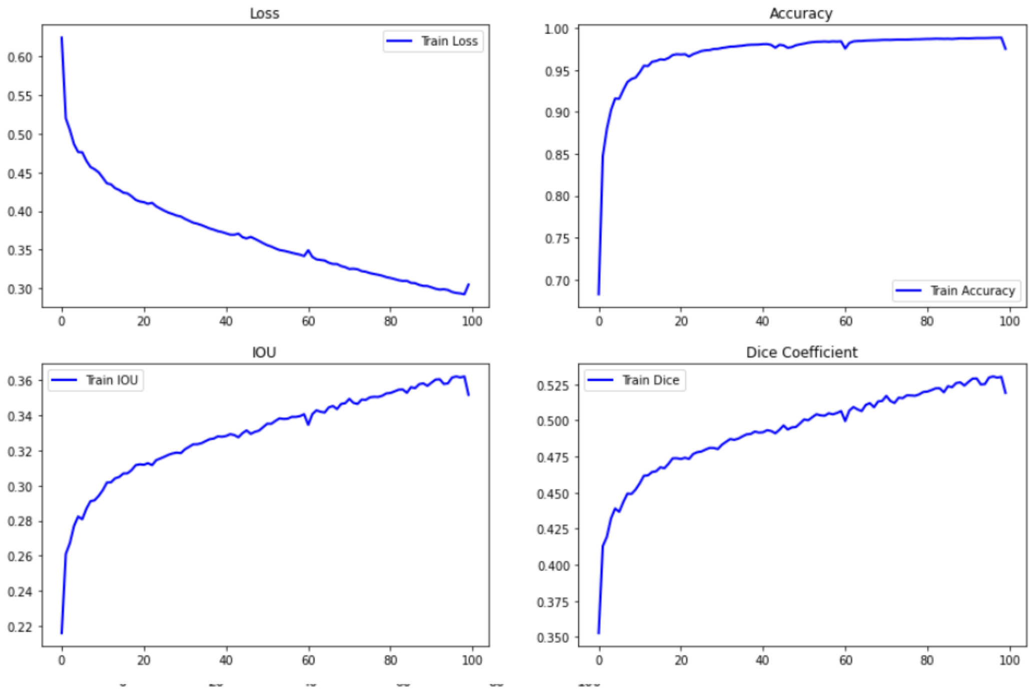


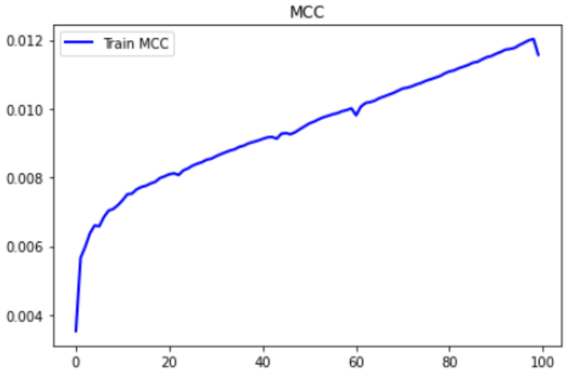
```
Image number: 2
```

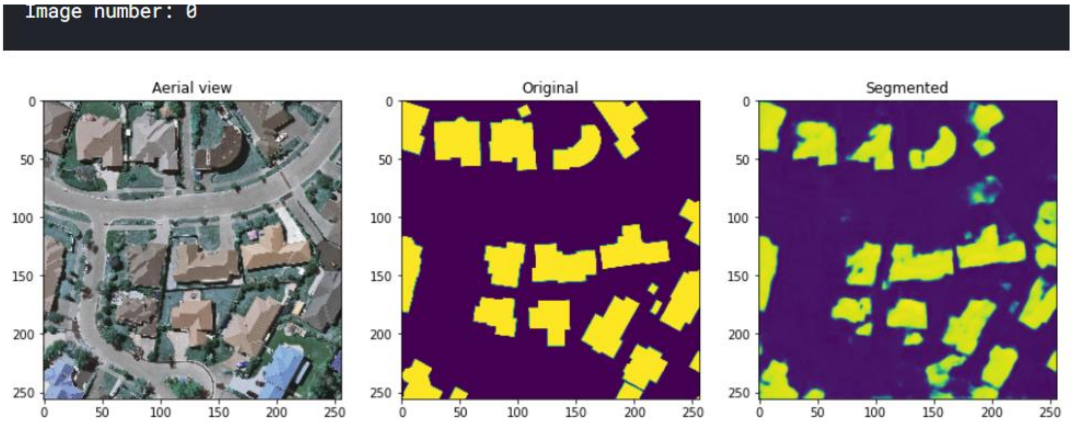


```
Image number: 3
```

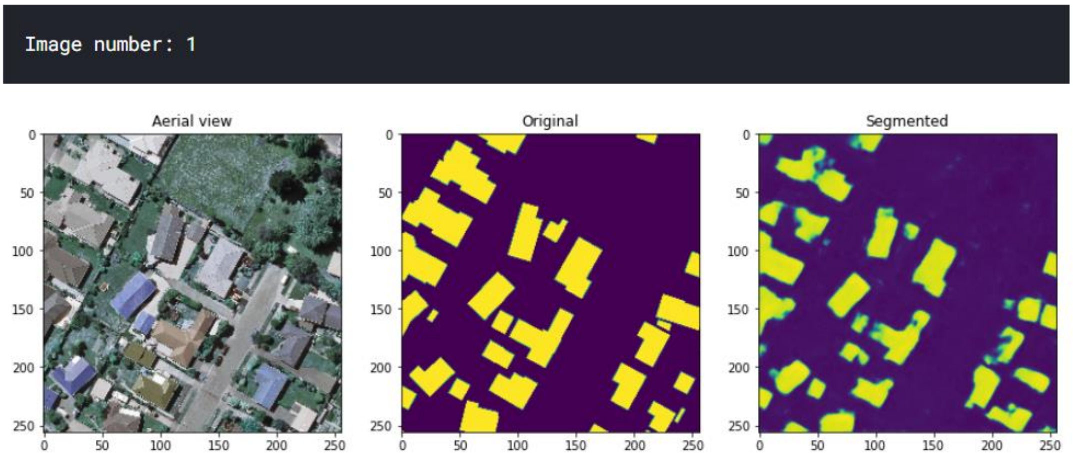
Model Name: MultiRes-100epochs-lr1e-4
Architecture: MultiRes U-Net
Numpy or Image Generator: Numpy
Epochs: 100
Batch size: 16
Learning rate: 0.0001
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Kaggle
Filename: Keras Data Generator Version 1 - multires-30
Results: IOU - 35.18, Dice Coefficient - 51.91, MCC - 1.66, Accuracy - 97.51, Loss - 0.3047
Comments: IOU with MultiRes doesn't increase after 35% even with 100 epochs for different learning rates as well. For learning rate = 0.01, IOU at end of 100 epochs is 38.93%.



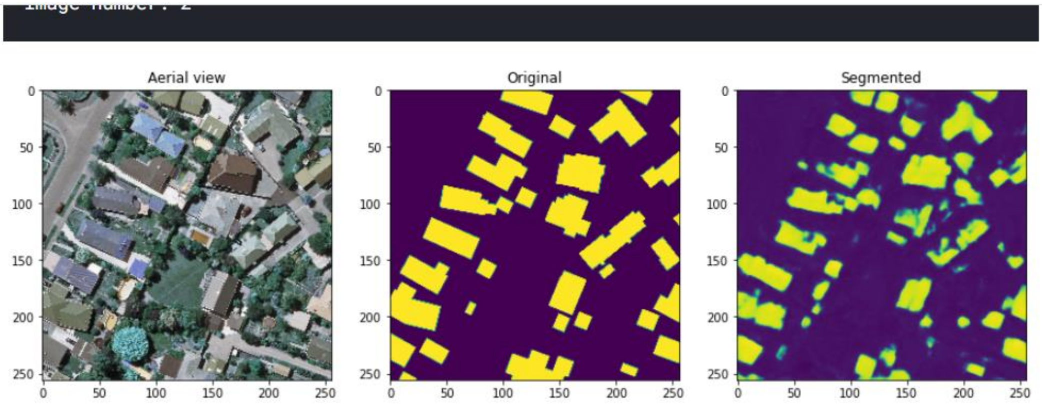




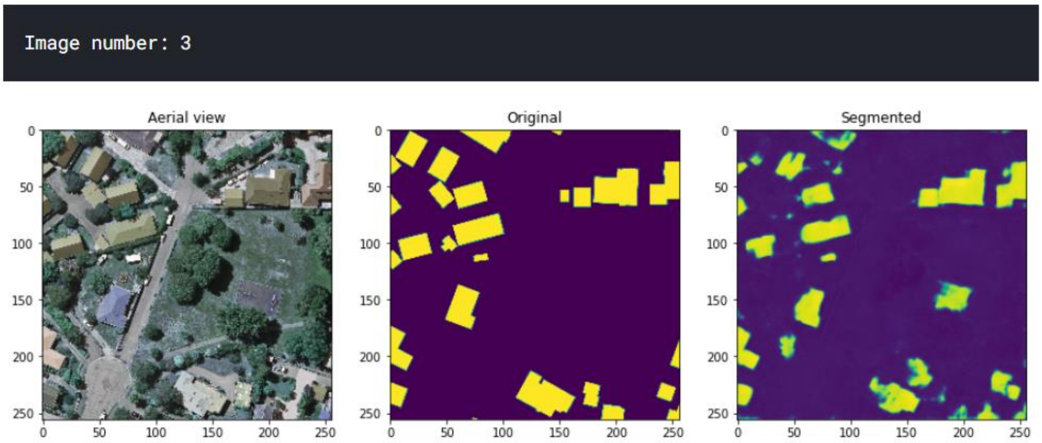
```
Image number: 0
```



```
Image number: 1
```

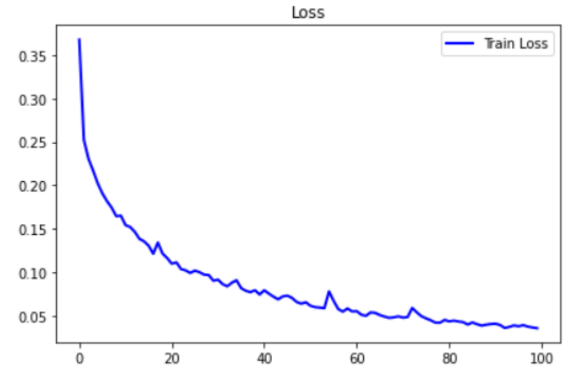


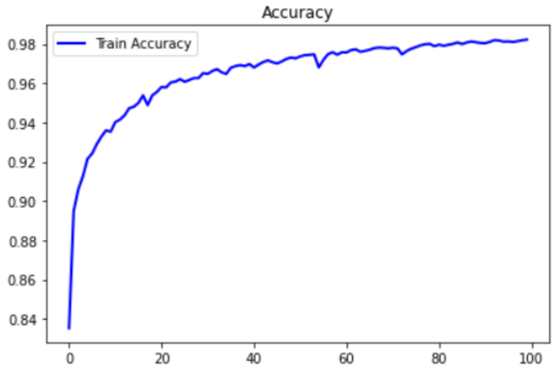
```
Image number: 2
```

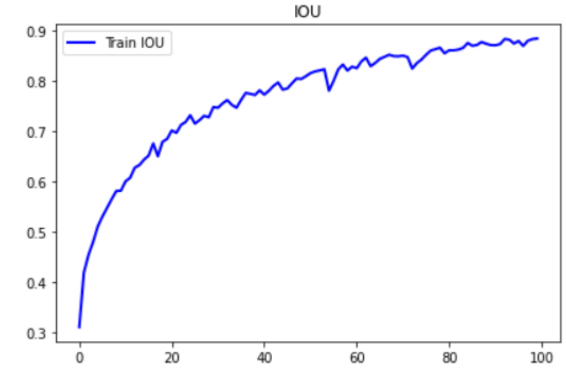


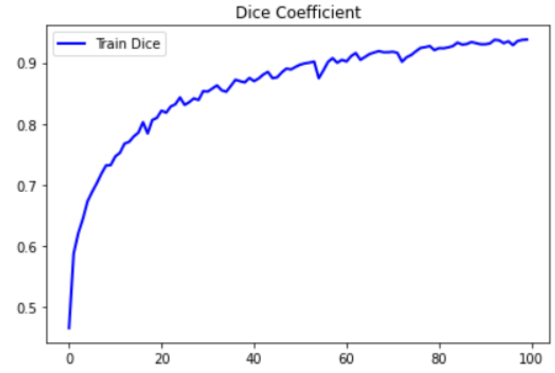
```
Image number: 3
```

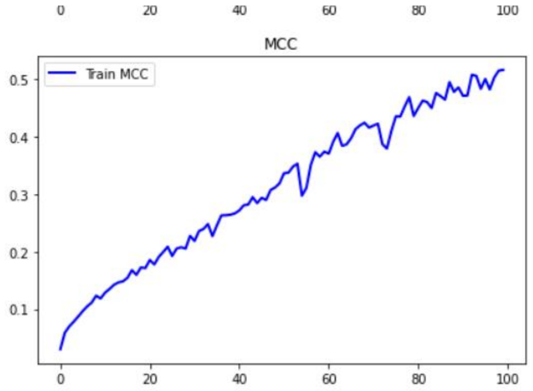
Model Name: unet2-1000images-100epochs
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 100
Batch size: 8
Learning rate: 0.0001
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Colab
Filename: U-Net Final
Results: IOU - 88.40, Dice Coefficient - 93.79, MCC – 51.68, Accuracy – 98.22, Loss – 0.0356
Comments: Tried resumable models too but that wasn't done properly.

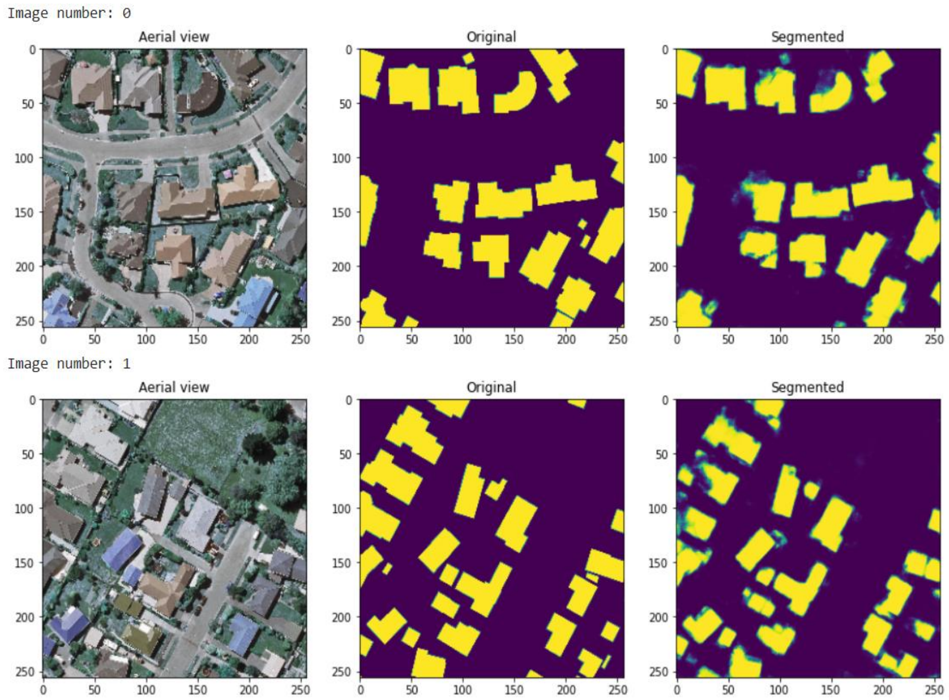






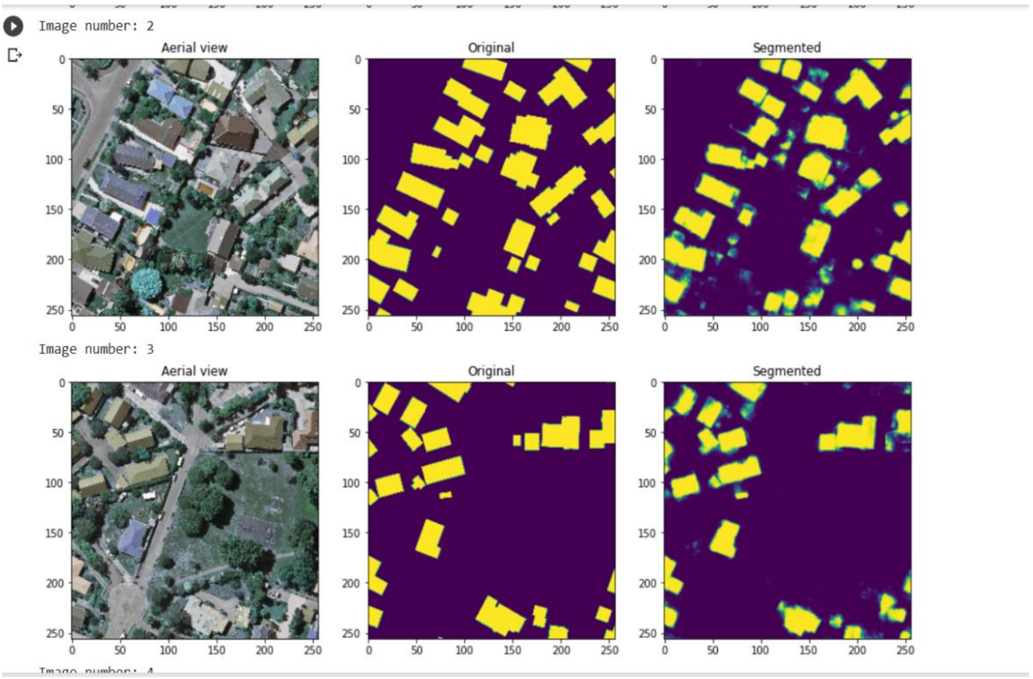






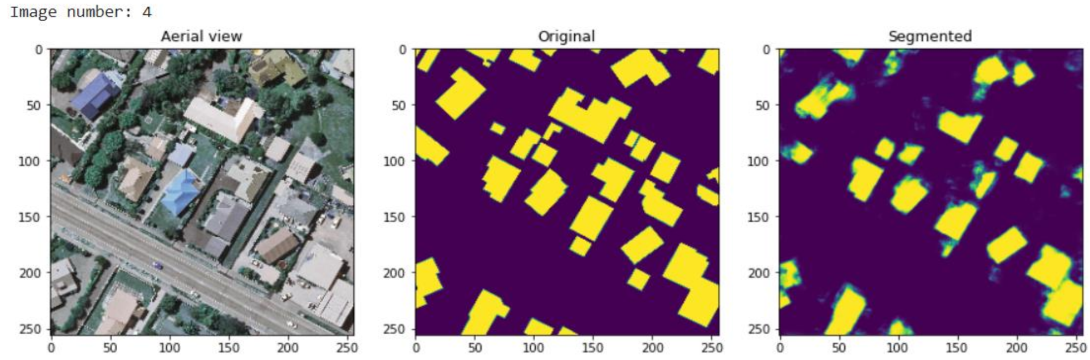
```
Image number: 0
```

```
Image number: 1
```



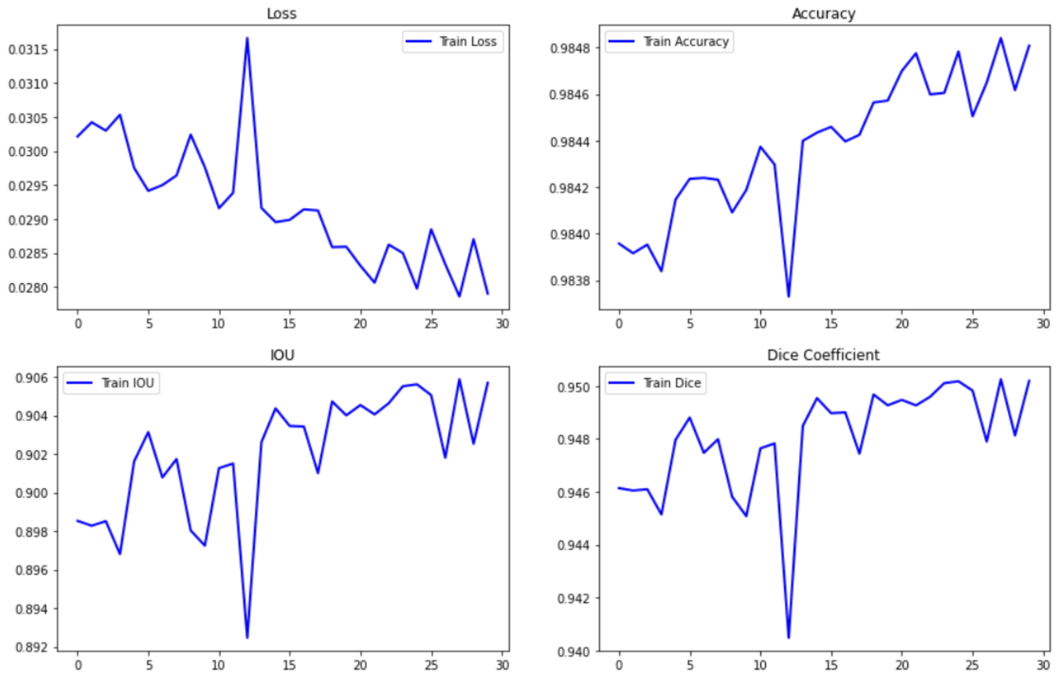
```
Image number: 2
```

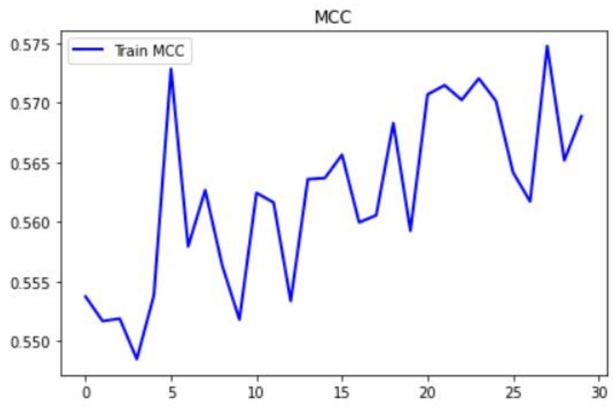
```
Image number: 3
```



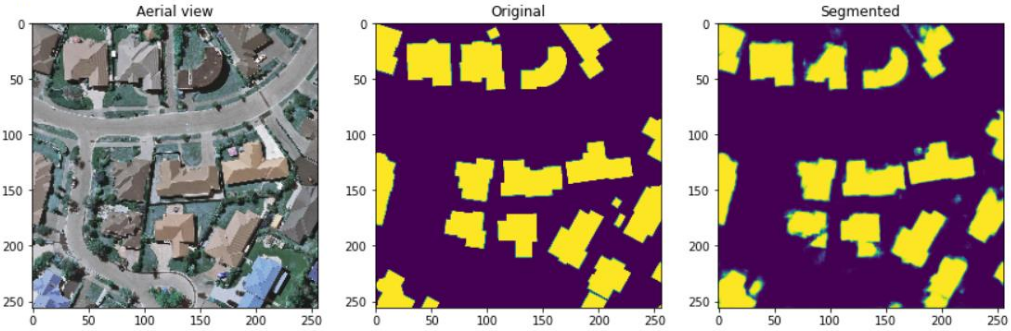
```
Image number: 4
```

Model Name: unet2-1000images-130epochs
Architecture: U-Net
Numpy or Image Generator: Numpy
Epochs: 30 (Continue training from the previous model) 100+30=130
Batch size: 8
Learning rate: 0.00001
No of images: 1008
Performance Metrics: IOU, Dice Coefficient, MCC, Accuracy
Colab or Kaggle: Colab
Filename: U-Net Final
Results: IOU - 90.57, Dice Coefficient - 95.02, MCC - 56.88, Accuracy - 98.48, Loss - 0.0279
Comments: Continued training from the previous model by loading the weights and decreasing the learning rate. Trained for 30 more epochs.

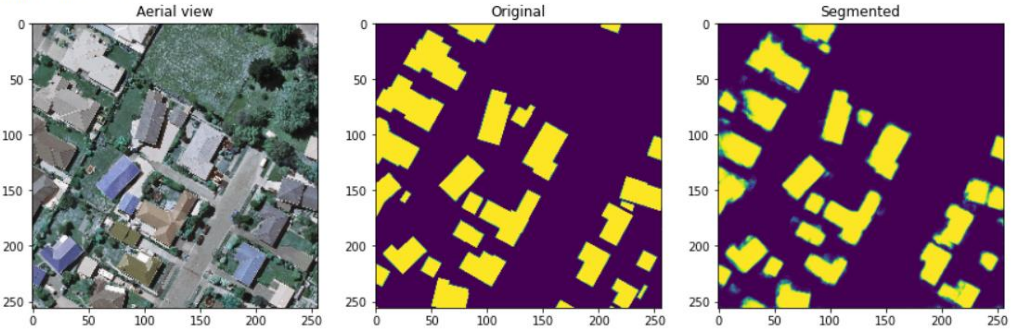




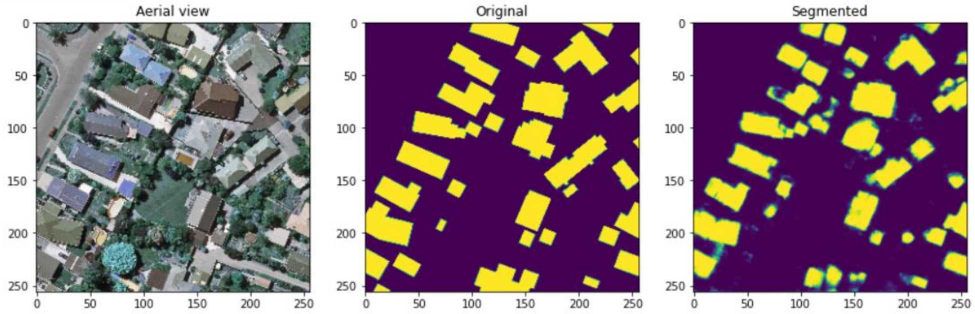
```
Image number: 0
```



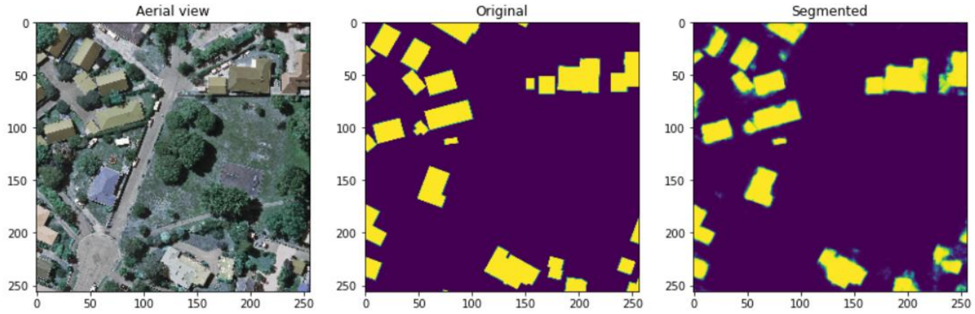
```
Image number: 1
```



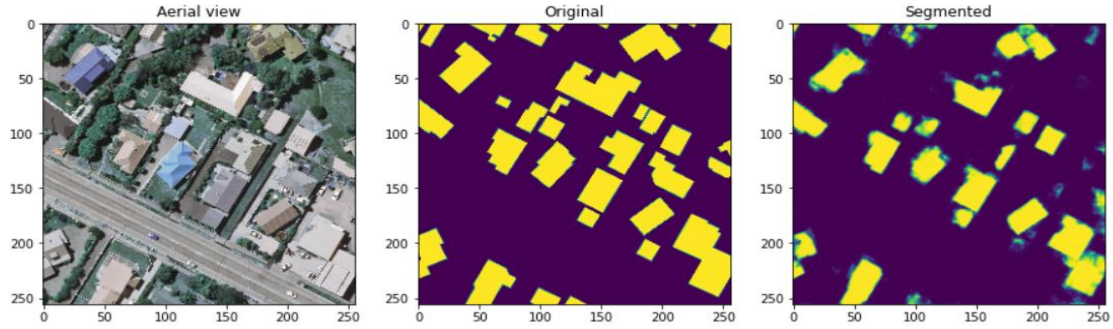
```
Image number: 2
```



```
Image number: 3
```



```
Image number: 4
```


Aerial view
Original
Segmented
0
50
100
150
200
250