PROJECT ALETHEIA

THE ONES N ZEROS

NM391 The Ones n Zeros

MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Dr. Vishwanath Karad

MIT WORLD PEACE UNIVERSITY | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SIH

SMART INDIA HACKATHON 2020
Finale

Organization Name :Indian Space Research Organization (ISRO)

Problem Statement: Reconstruction of missing data in Satellite Imagery

PS Number : NM391

Team Name : The Ones n Zeros

Team Leader Name : Vignesh Charan Raman

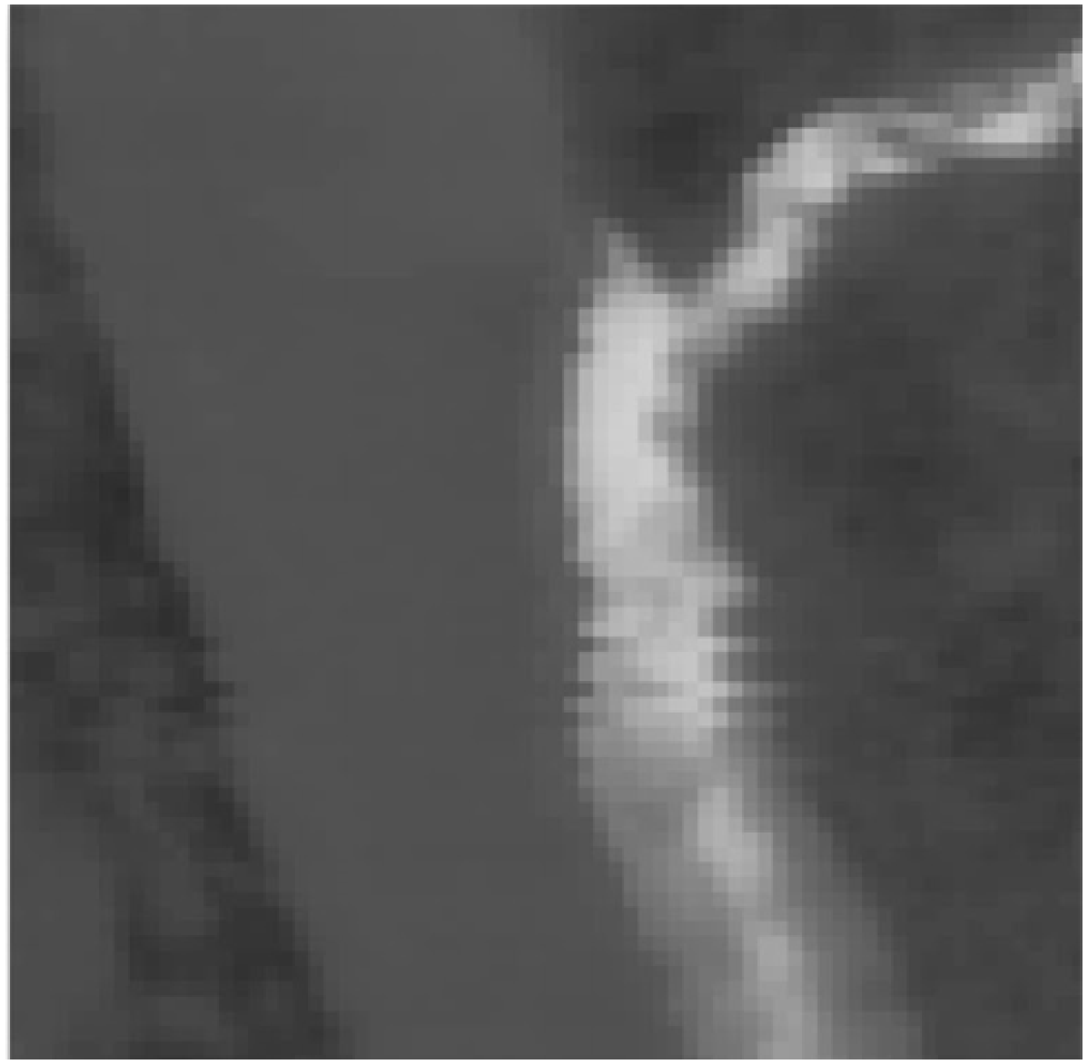
College code : 1-3713389121

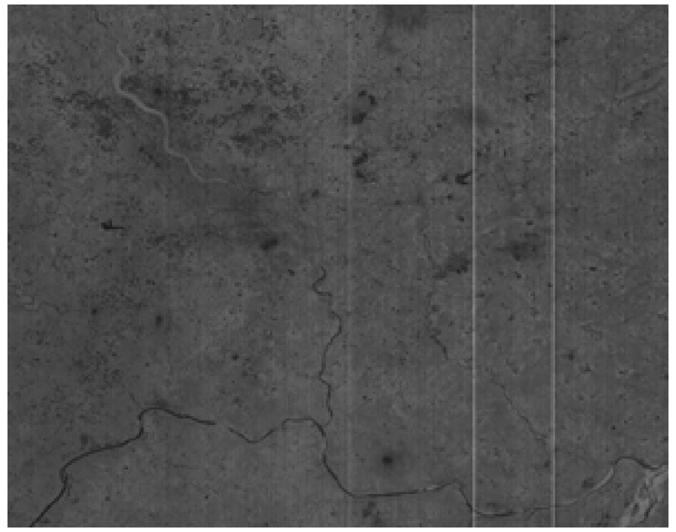
Problem Statment

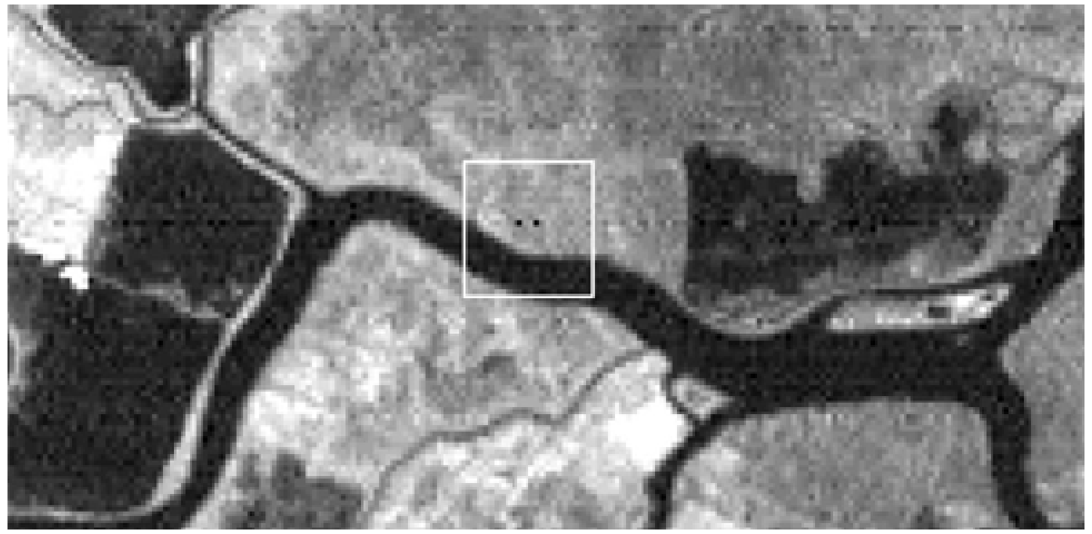
NM391-ISRO Reconstruction of missing data in Satellite Imagery

Short Wave Infra-Red(SWIR) detectors used in satellite imaging cameras suffer from dropouts in pixel and line direction in raw data. Develop software to reconstruct missing parts of a satellite image so that observers are unable to identify regions that have undergone reconstruction

TYPES OF ERRORS

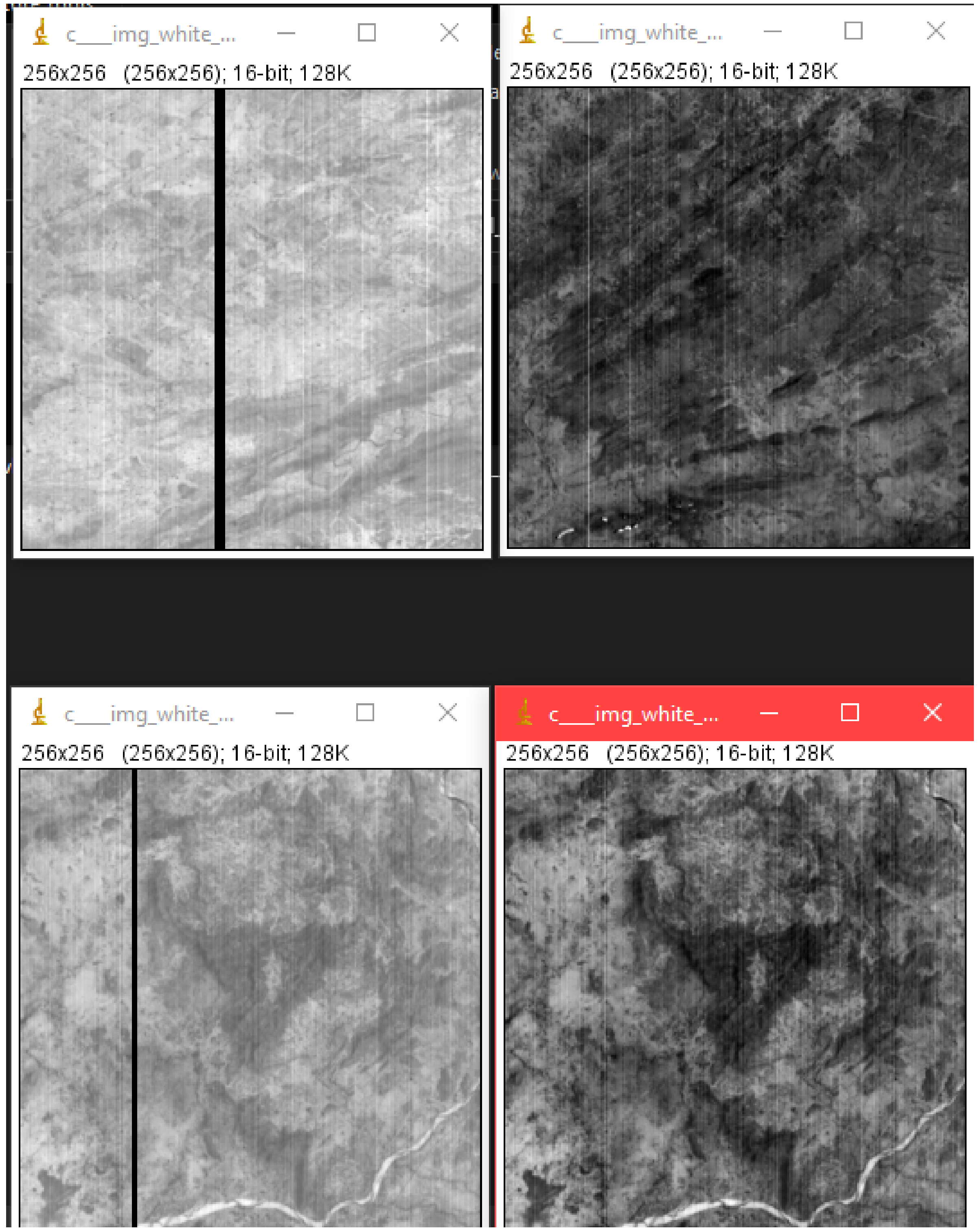






- Random bad pixels (shot noise).
- Line-start/stop problems.
- Line or column drop-outs.
- Partial line or column drop-outs.
- Line or column striping

Proposed Solution



The method we intend to implement relies on the algorithm namely DC-GANs to Reconstruct missing parts of images effectively.

The Training Dataset provided to the Algorithm learns the correlation between the pixels of the image. The nearby pixels are analysed (spatial analysis) and the correlation is applied to fill the incomplete data and the most homologous result which is the final reconstructed image is given as an output.

Since the proposed algorithm uses pristine analysis, the reconstructed Image promises high accuracy with almost no room for error.

Architechture of GAN

```
x = gen_conv(x, cnum, 5, 1, name='conv1')
x = gen_conv(x, 2*cnum, 3, 2, name='conv2_downsample')
x = gen_conv(x, 2*cnum, 3, 1, name='conv3')
x = gen_conv(x, 4*cnum, 3, 2, name='conv4_downsample')
x = gen_conv(x, 4*cnum, 3, 1, name='conv5')
x = gen_conv(x, 4*cnum, 3, 1, name='conv6')
mask_s = resize_mask_like(mask, x)
x = gen_conv(x, 4*cnum, 3, rate=2, name='conv7_atrous')
x = gen_conv(x, 4*cnum, 3, rate=4, name='conv8_atrous')
x = gen_conv(x, 4*cnum, 3, rate=8, name='conv9_atrous')
x = gen_conv(x, 4*cnum, 3, rate=16, name='conv10_atrous')
x = gen_conv(x, 4*cnum, 3, 1, name='conv11')
x = gen_conv(x, 4*cnum, 3, 1, name='conv12')
x = gen_deconv(x, 2*cnum, name='conv13_upsample')
x = gen_conv(x, 2*cnum, 3, 1, name='conv14')
x = gen_deconv(x, cnum, name='conv15_upsample')
x = gen_conv(x, cnum//2, 3, 1, name='conv16')
x = gen_conv(x, 3, 3, 1, activation=None, name='conv17')
x = tf.nn.tanh(x)
x_stage1 = x
```

Generator Stage 1

Normal Feed forward NN With dialated Convolutional Layers (used for expanding the kernels to extract the features nearby the reconstructing regions)

Architechture of GAN

```
x = x*mask + xin[:, :, :, 0:3]*(1.-mask)
x.set_shape(xin[:, :, :, 0:3].get_shape().as_list())

xnow = x
x = gen_conv(xnow, cnum, 5, 1, name='xconv1')
x = gen_conv(x, cnum, 3, 2, name='xconv2_downsample')
x = gen_conv(x, 2*cnum, 3, 1, name='xconv3')
x = gen_conv(x, 2*cnum, 3, 2, name='xconv4_downsample')
x = gen_conv(x, 4*cnum, 3, 1, name='xconv5')
x = gen_conv(x, 4*cnum, 3, 1, name='xconv6')
x = gen_conv(x, 4*cnum, 3, rate=2, name='xconv7_atrous')
x = gen_conv(x, 4*cnum, 3, rate=4, name='xconv8_atrous')
x = gen_conv(x, 4*cnum, 3, rate=8, name='xconv9_atrous')
x = gen_conv(x, 4*cnum, 3, rate=16, name='xconv10_atrous')
```

Generator   Stage 2

Attention Branch

Contextual Attention or making the algorithm context aware.
The features extracted from dialation layer are used here for making the algorithm analyze the regions.

Architechture of GAN

```
x = gen_conv(xnow, cnum, 5, 1, name='pmconv1')
x = gen_conv(x, cnum, 3, 2, name='pmconv2_downsample')
x = gen_conv(x, 2*cnum, 3, 1, name='pmconv3')
x = gen_conv(x, 4*cnum, 3, 2, name='pmconv4_downsample')
x = gen_conv(x, 4*cnum, 3, 1, name='pmconv5')
x = gen_conv(x, 4*cnum, 3, 1, name='pmconv6',
                    activation=tf.nn.relu)
x, offset_flow = contextual_attention(x, x, mask_s, 3, 1, rate=2)
x = gen_conv(x, 4*cnum, 3, 1, name='pmconv9')
x = gen_conv(x, 4*cnum, 3, 1, name='pmconv10')
pm = x
x = tf.concat([x_hallu, pm], axis=3)

x = gen_conv(x, 4*cnum, 3, 1, name='allconv11')
x = gen_conv(x, 4*cnum, 3, 1, name='allconv12')
x = gen_deconv(x, 2*cnum, name='allconv13_upsample')
x = gen_conv(x, 2*cnum, 3, 1, name='allconv14')
x = gen_deconv(x, cnum, name='allconv15_upsample')
x = gen_conv(x, cnum//2, 3, 1, name='allconv16')
x = gen_conv(x, 3, 3, 1, activation=None, name='allconv17')
x = tf.nn.tanh(x)
x_stage2 = x
```

Generator Stage 2

All Layers are aggregated and fed into single decoder for obtaining the output. deconvolution to reconstruct the generated patches with contextual patches.

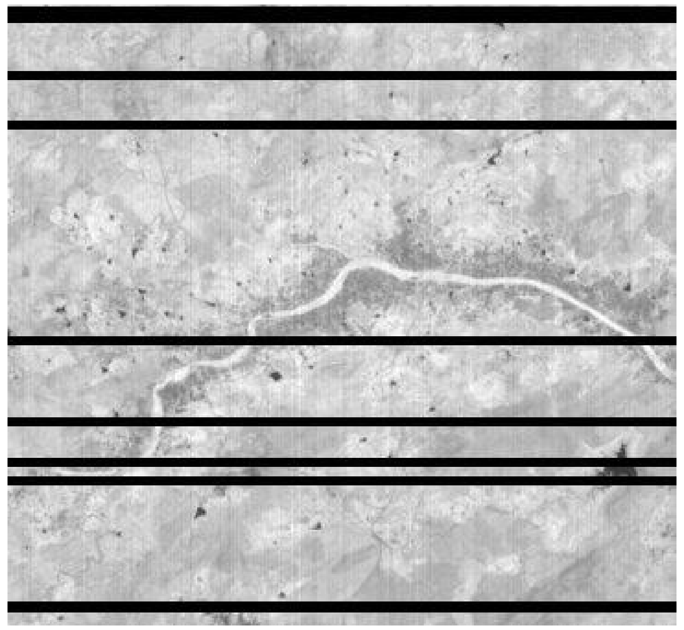
Architechture of GAN

```
x = dis_conv(x, cnum, name='conv1', training=training)
x = dis_conv(x, cnum*2, name='conv2', training=training)
x = dis_conv(x, cnum*4, name='conv3', training=training)
x = dis_conv(x, cnum*4, name='conv4', training=training)
x = dis_conv(x, cnum*4, name='conv5', training=training)
x = dis_conv(x, cnum*4, name='conv6', training=training)
x = flatten(x, name='flatten')
```

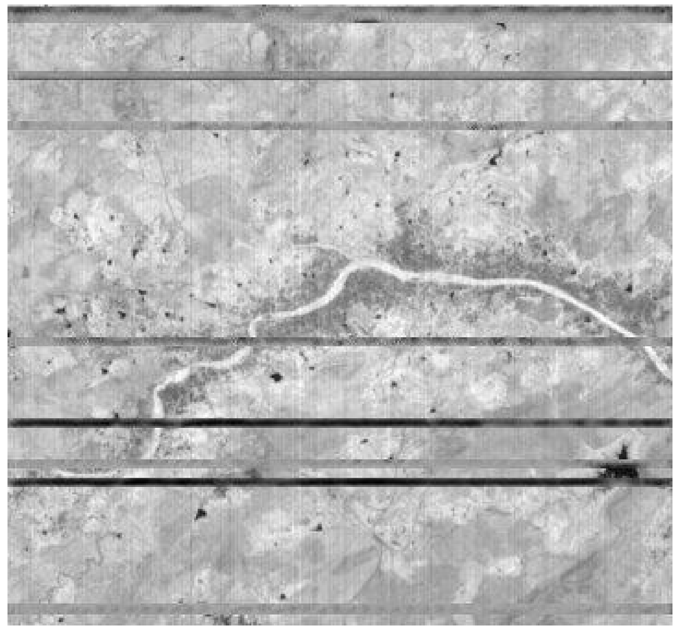
Discriminator

A Normal Discriminator is attached to the 2 staged Generator

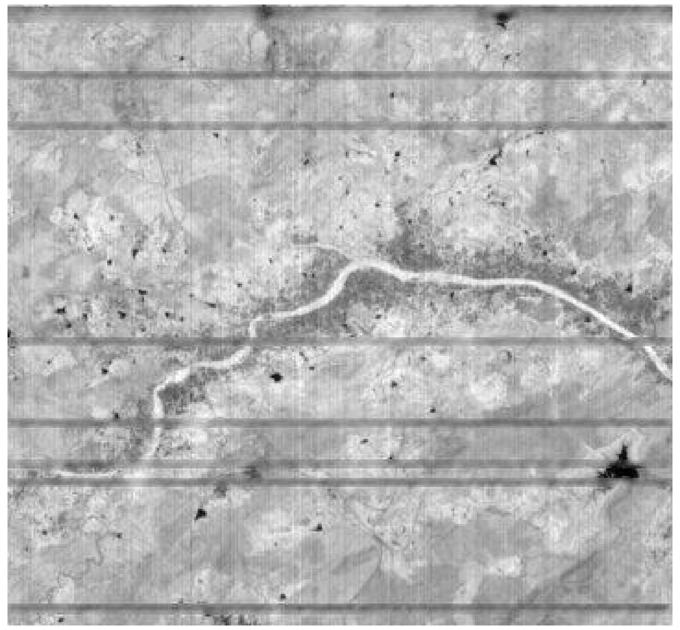
Output



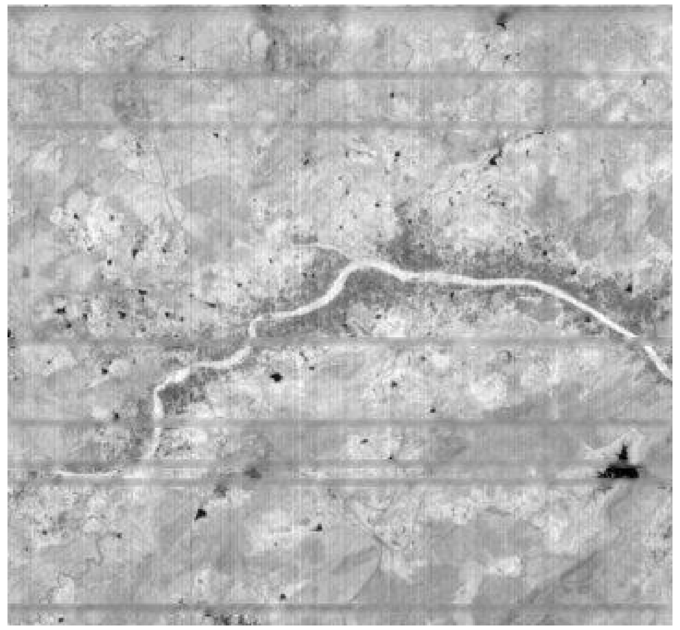
Input



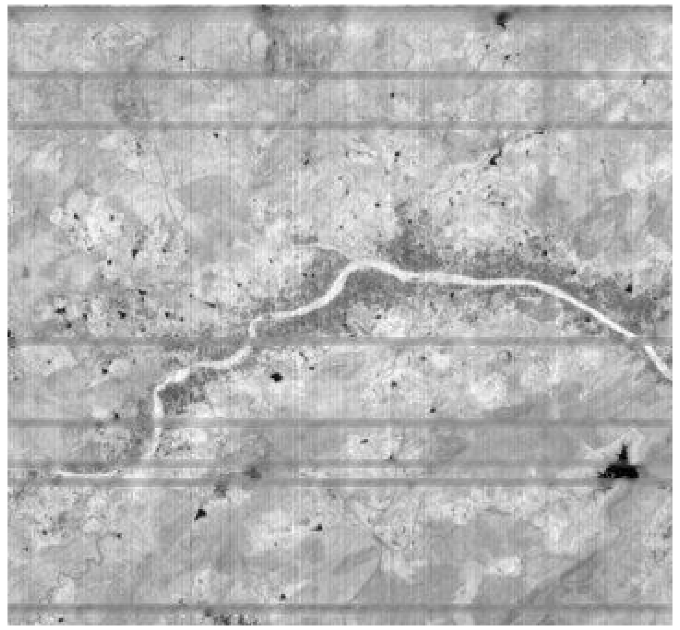
Output from Gans 1st generation (61epoches)



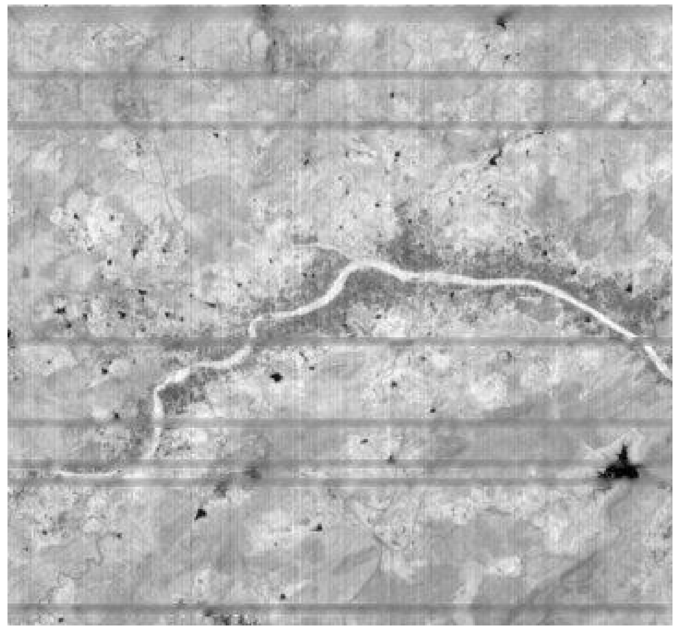
Output from New Gans model 2 staged (82 epoches)



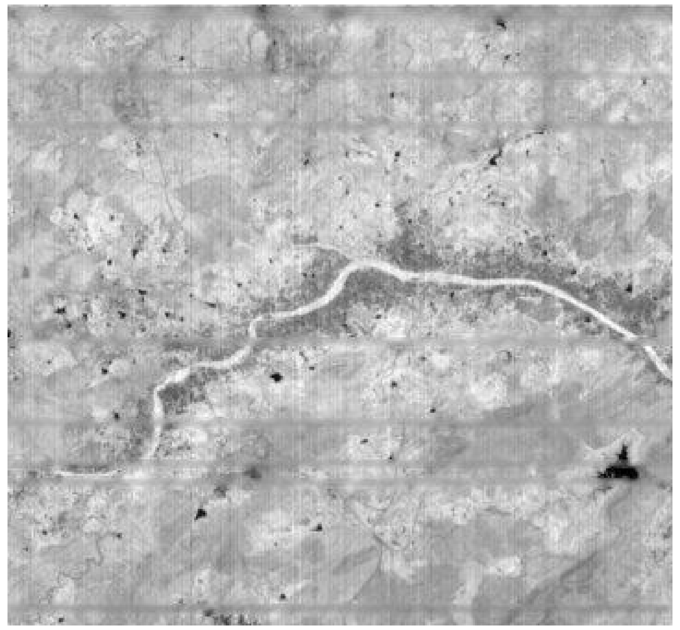
New Gans (118 epoches)



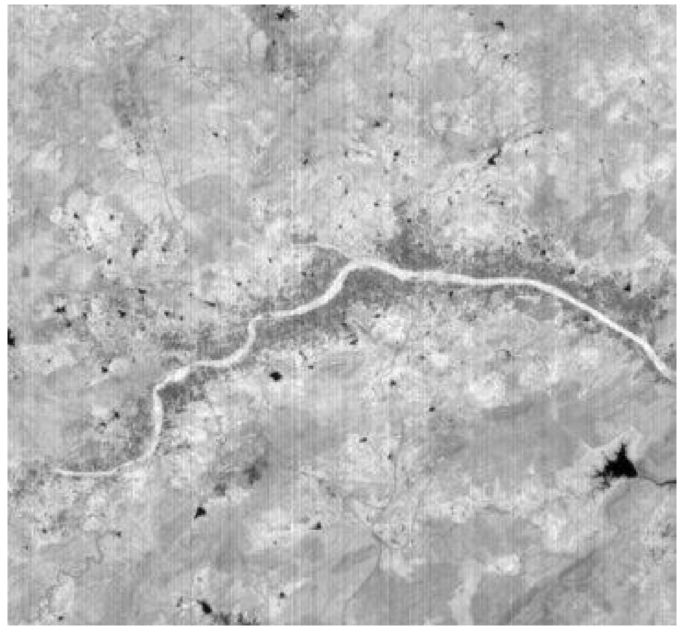
New Gans (201epoches)



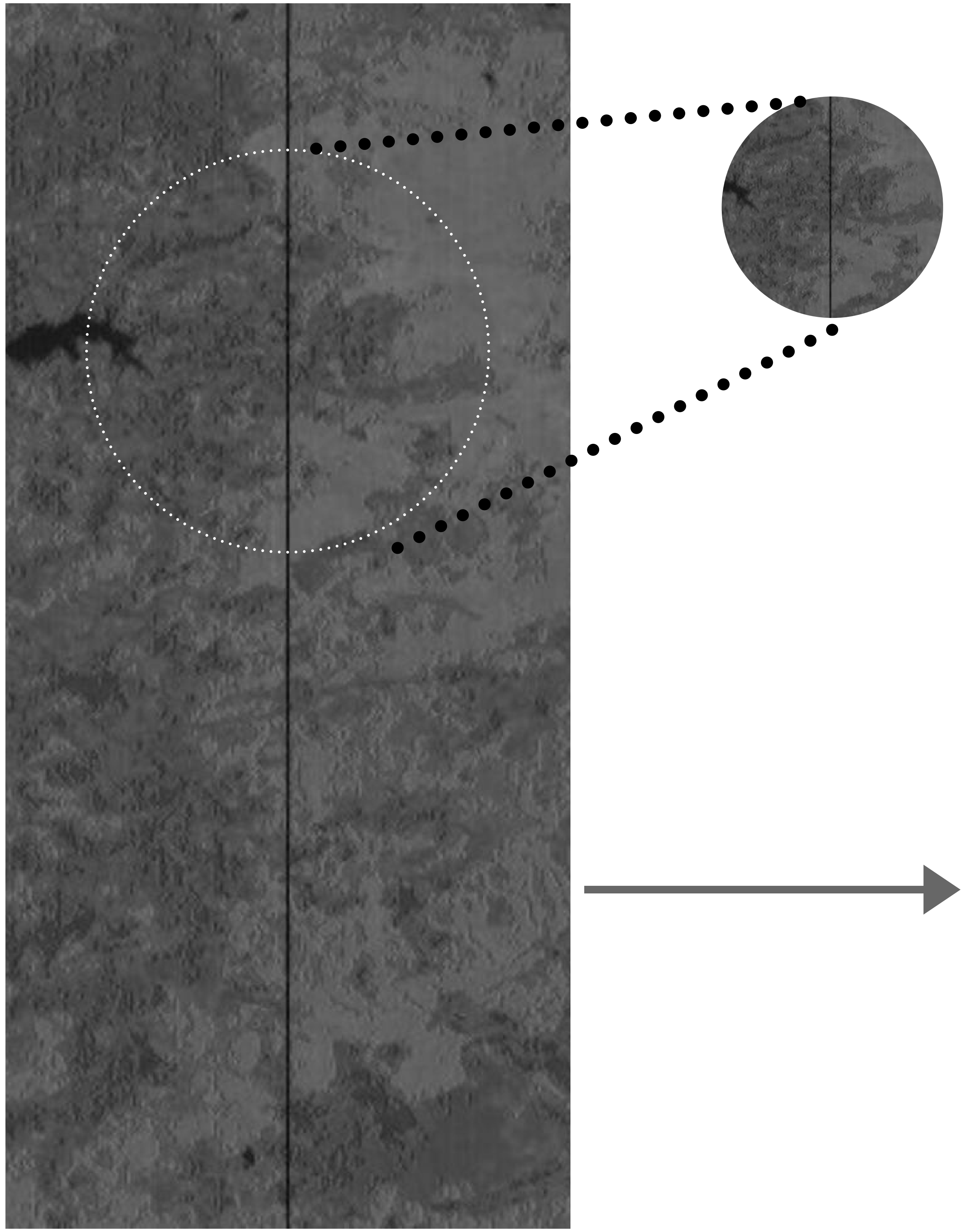
New Gans (237epoches)



New Gans (359 epoches)

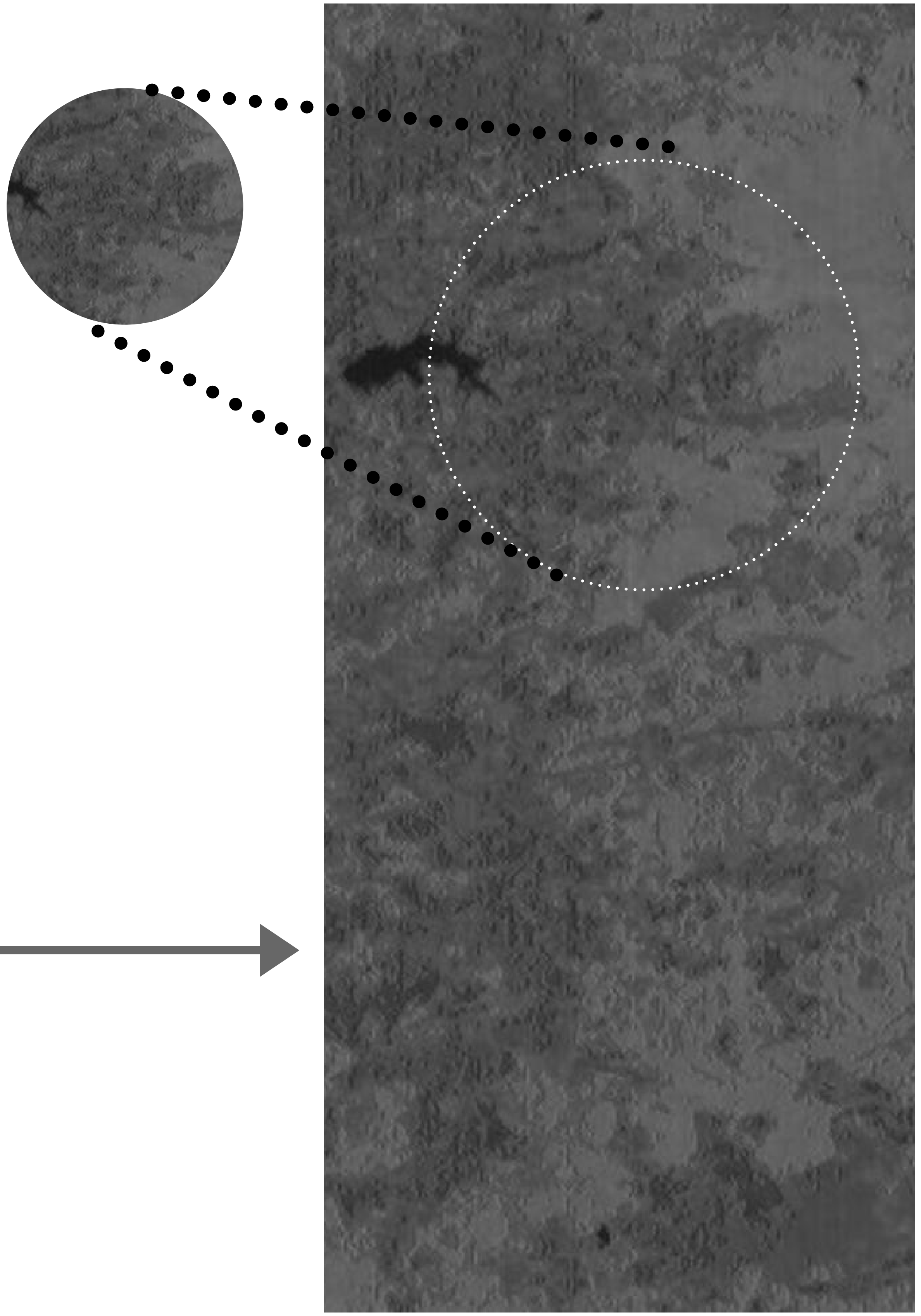


Original Ground Truth Image



Generative Inpainting

By using 2 Staged GAN algorithm which is robust to calculate the pixel values of missing regions.



Results

Similarity Index - SSIM values :

1. Original vs input : **0.9314753871768957**
2. Original vs New Gans algo with new training data (359 epoches) : **0.9996847497074818**
3. Original vs New Gans algo with new training data (82 epoches) : **0.9986725530035611**
4. Original vs Old Gans algo with old training data : **0.9942820439040533**
5. Original vs OpenCV reconstructed output : **0.9963796319465636**

Citations

Title= Free-Form Image Inpainting with Gated Convolution
author=Yu, Jiahui and Lin, Zhe and Yang, Jimei and Shen, Xiaohui and Lu, Xin and Huang, Thomas S
journal=arXiv preprint arXiv:1806.03589

THANK YOU