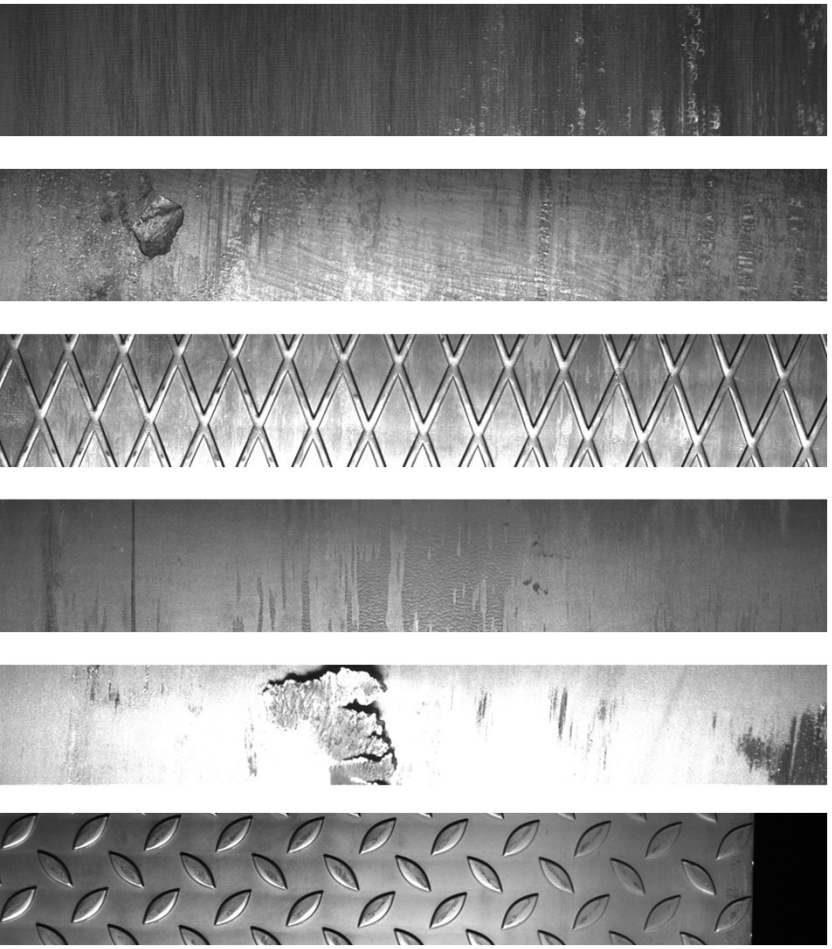
Steel Slab Damage Marking



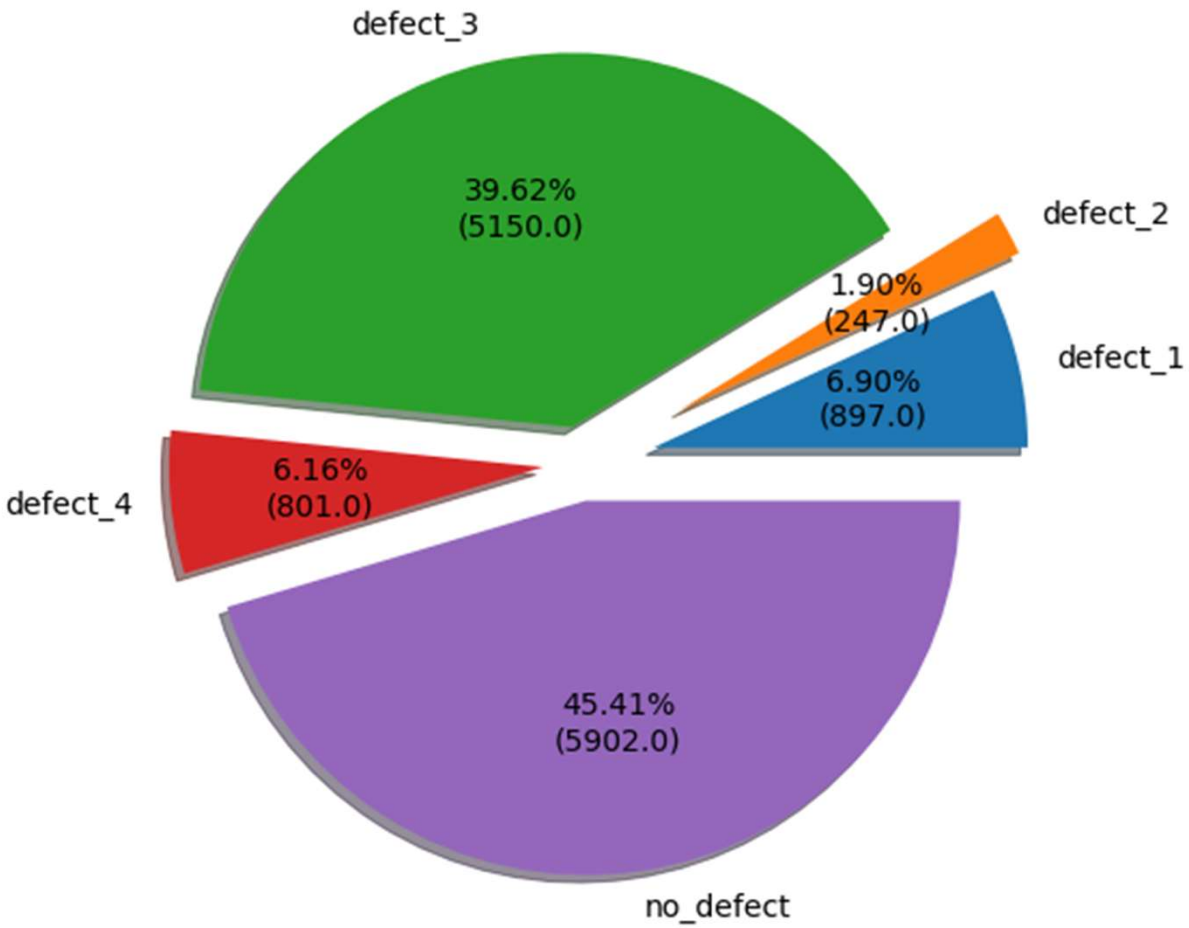
- Total categories: No_defect, defect_1, defect_2, defect_3, defect_4

- Classes in focus: defect_1, defect_2, defect_3, defect_4

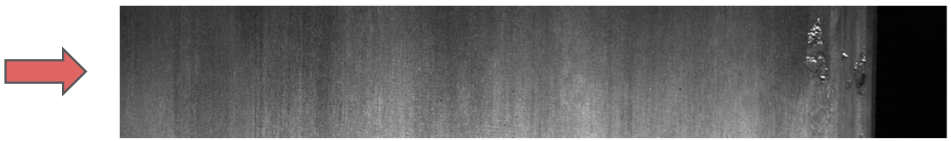
- Total: 18,074 images and corresponding damage

Encoded pixels.

Dataset classification chart.



Defect - Pixel Encoding



RGB image is (256,1600,3) 256*1600 = 409,600 locations

Example image *01661826d.jpg*

```
339049 1 339291 8 339305 3 339541 25 339789 34 340038 43 340290
53 340544 60 340798 63 341052 67 341306 70 341559 74 341811 79
342065 82 342319 85 342574 87 342828 90 343082 92 343337 93
343591 95 343846 95 344100 97 344355 98 344609 100 344864 101
345118 103 345372 104 345626 106
```

First number is starting pixel and next number is count from that starting pixel.

Defect - Pixel Encoding

Example image *01661826d.jpg* 

`339049 1 339291 8 339305 3`

`339049 1` – 1 pixel starting from 339049

`339291 8` – 8 pixels starting from 339291

`339305 3` – 3 pixels starting from 339305

Defect - Pixel Encoding

Example image *01661826d.jpg*

`339049 1` – 1 pixel starting from 339049

`339291 8` – 8 pixels starting from 339291

`339305 3` – 3 pixels starting from 339305

Signifies:

339049	339291	339292	339293	339294	339295	339296	339297	339298	339305	339306	339307

Defective Pixel Encoding

Example image *01661826d.jpg*

```
339049 1 339291 8 339305 3 339541 25 339789 34 340038 43 340290 53
340544 60 340798 63 341052 67 341306 70 341559 74 341811 79 342065
82 342319 85 342574 87 342828 90 343082 92 343337 93 343591 95
343846 95 344100 97 344355 98 344609 100 344864 101 345118 103
345372 104 345626 106
```

Splitting the data frame into 2 dataframes; start_pixel and pixel_count

```
start_pixel = ['339049', '339291', '339305', '339541', '339789']
```

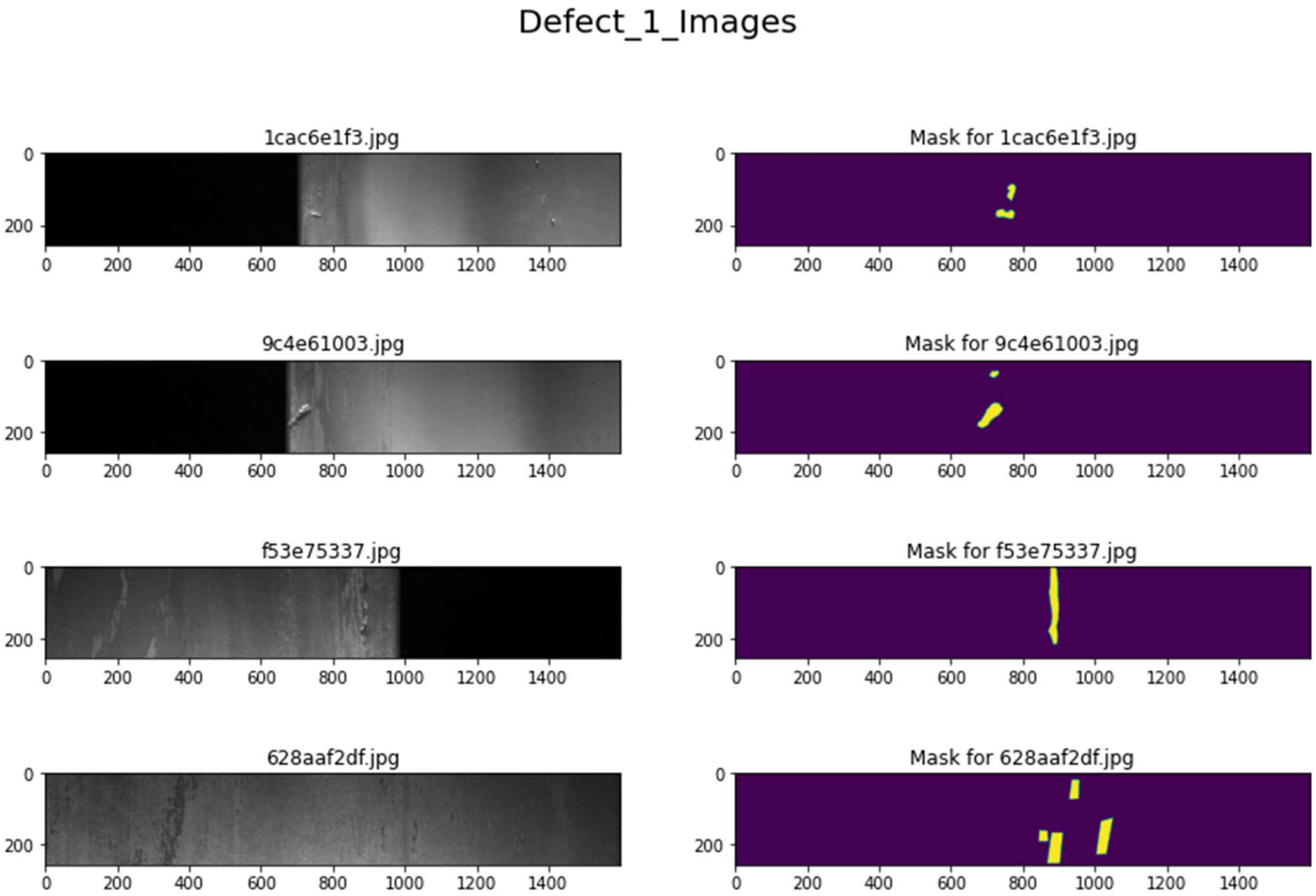
```
pixel_count = [ '1', '8', '3', '25', '34', '43', '53']
```

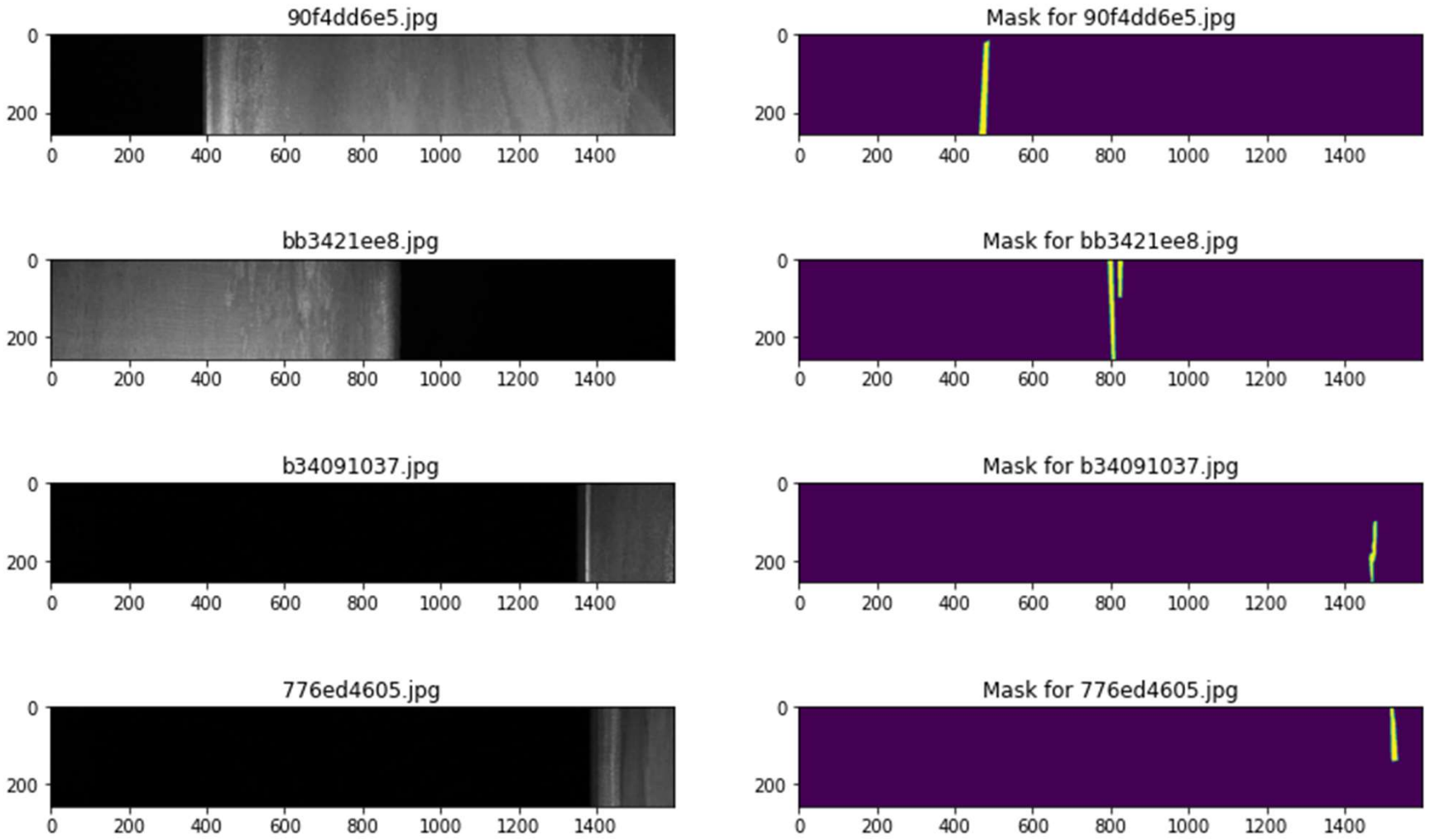
Defect - Pixel Encoding

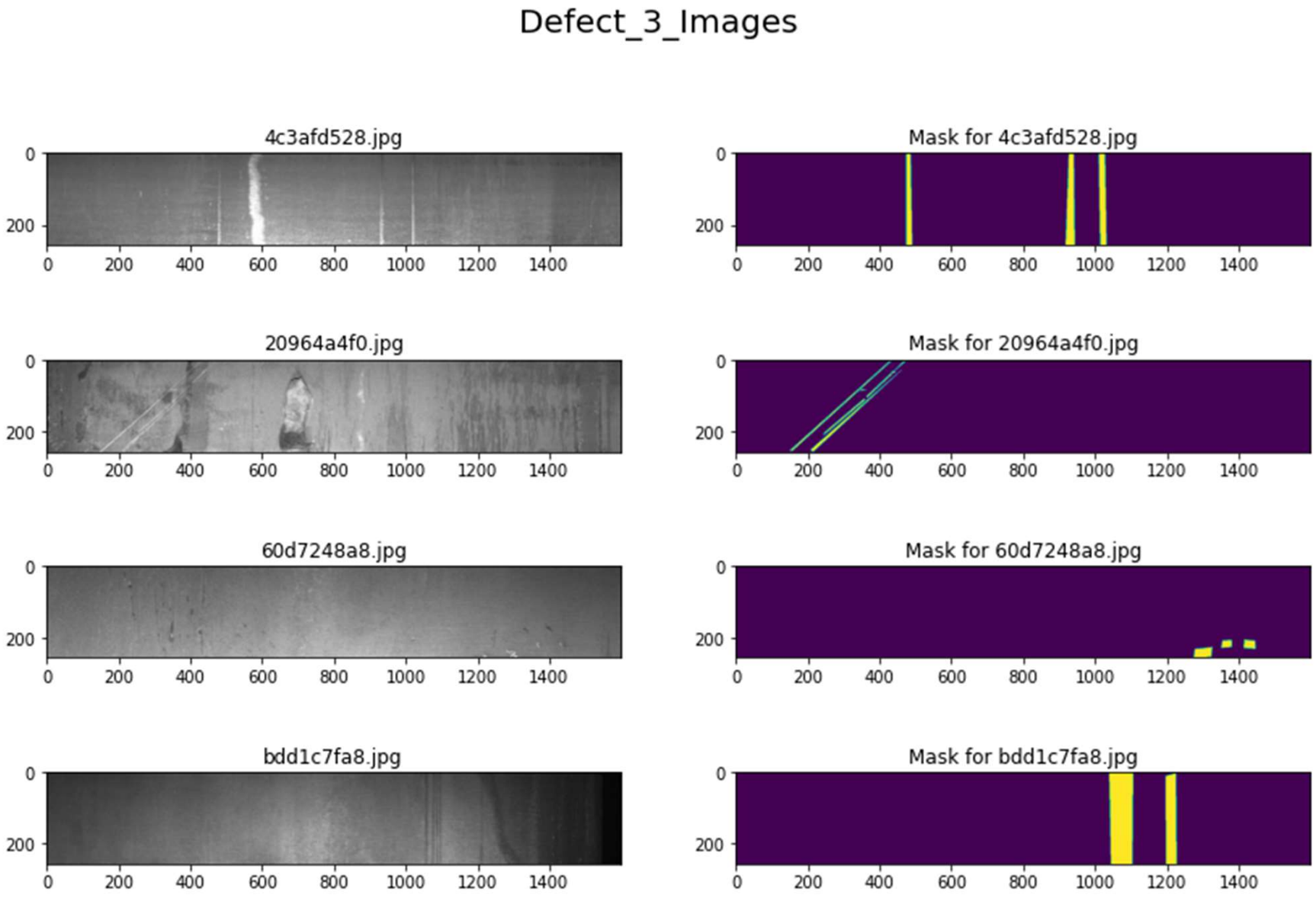
Example image *01661826d.jpg*

```
start_pixel = ['339049', '339291', '339305', '339541', '339789']
pixel_count = [ '1', '8', '3', '25', '34', '43', '53']

range(start_pixel[i],start_pixel[i] + pixel_count[i])


['339049', '339291', '339292', '339293', '339294', '339295', '339296',
'339297', '339298',....., '339841']
```

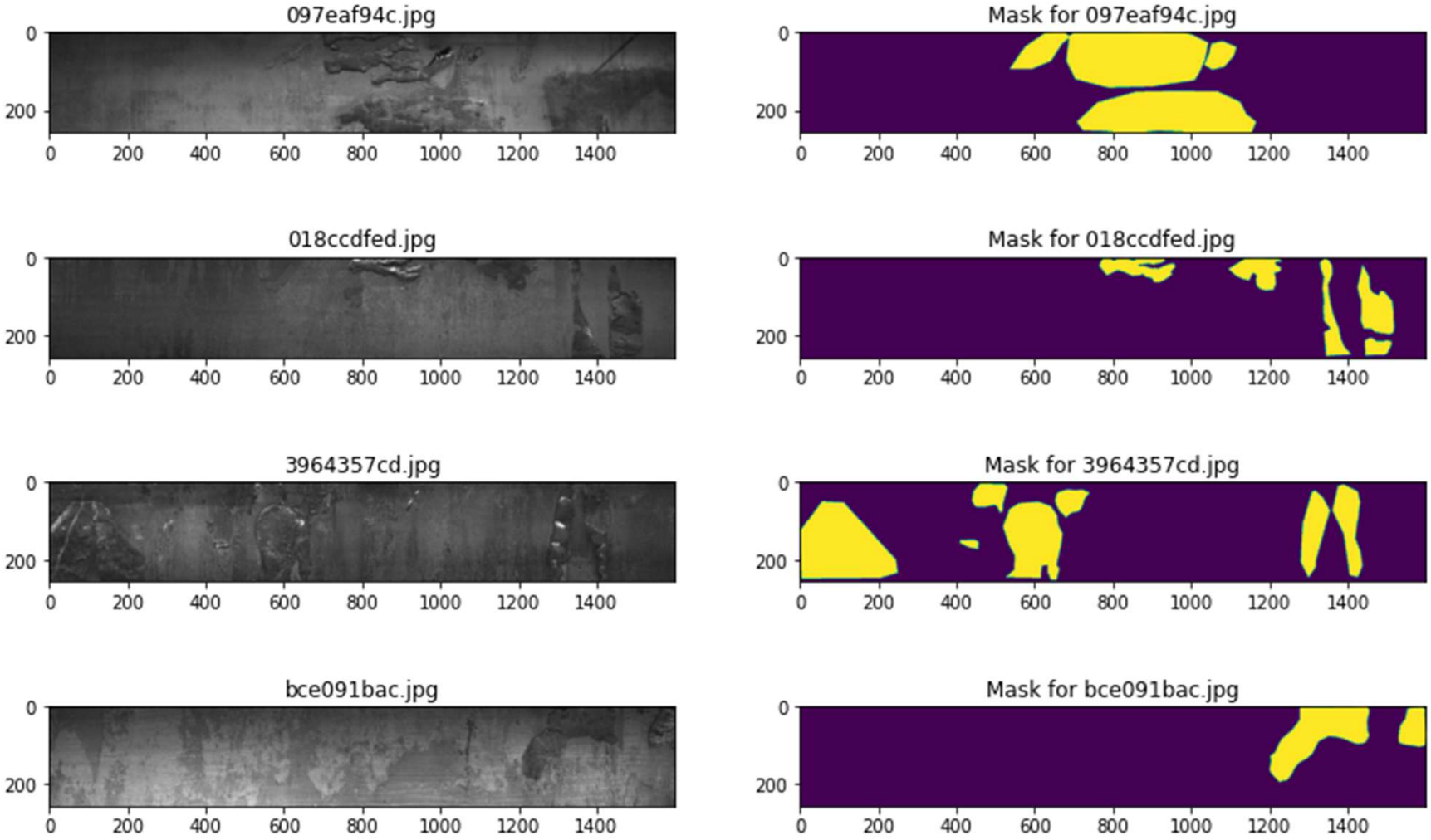
OpenCV for highlighting coordinates.


Defect_1_Images
1cac6e1f3.jpg
Mask for 1cac6e1f3.jpg
9c4e61003.jpg
Mask for 9c4e61003.jpg
f53e75337.jpg
Mask for f53e75337.jpg
628aaf2df.jpg
Mask for 628aaf2df.jpg
0
200
0
200
400
600
800
1000
1200
1400

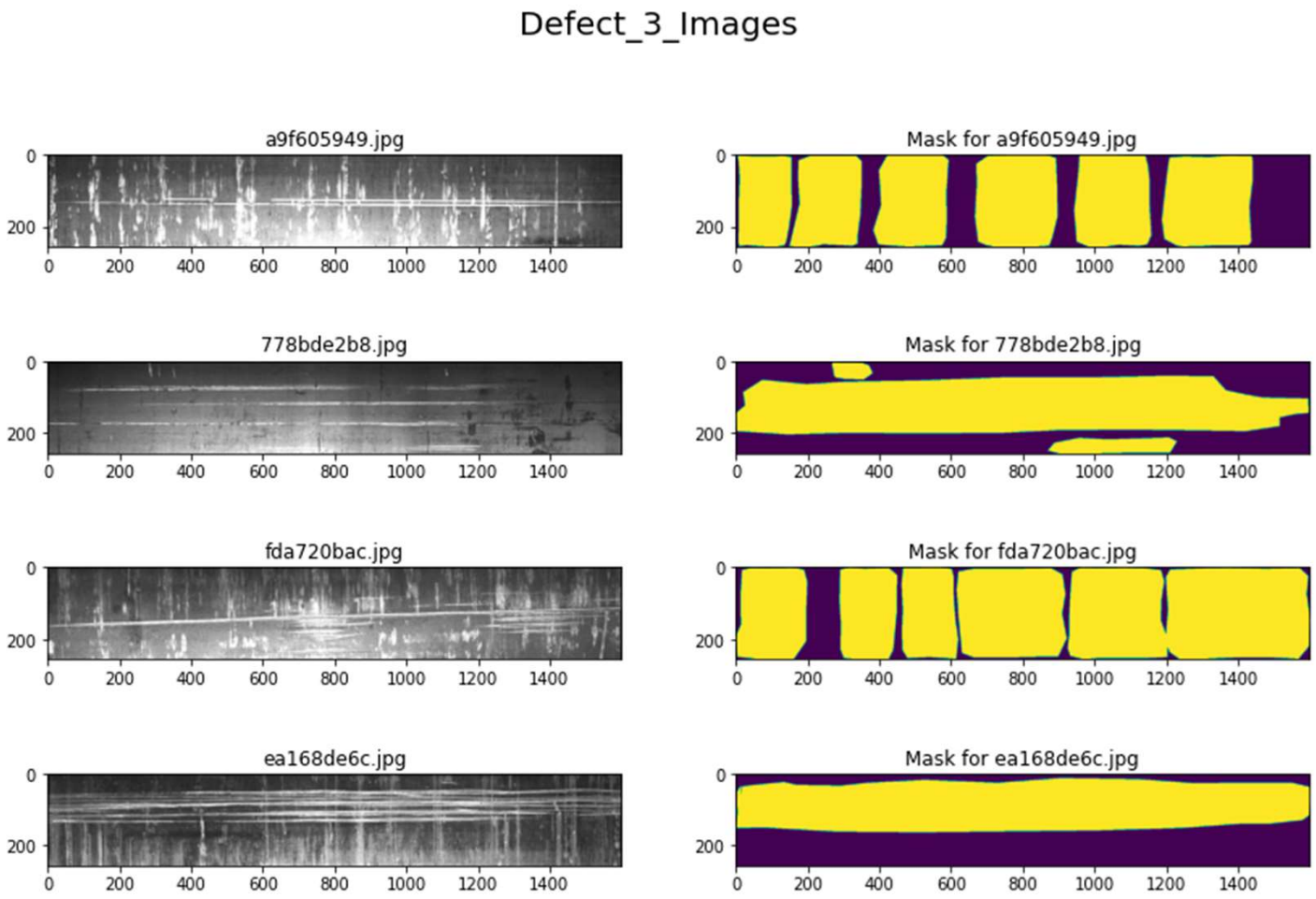

Defect_2_Images
90f4dd6e5.jpg
Mask for 90f4dd6e5.jpg
bb3421ee8.jpg
Mask for bb3421ee8.jpg
b34091037.jpg
Mask for b34091037.jpg
776ed4605.jpg
Mask for 776ed4605.jpg
0
200
0
200
400
600
800
1000
1200
1400


Defect_3_Images
4c3afd528.jpg
Mask for 4c3afd528.jpg
20964a4f0.jpg
Mask for 20964a4f0.jpg
60d7248a8.jpg
Mask for 60d7248a8.jpg
bdd1c7fa8.jpg
Mask for bdd1c7fa8.jpg
0
200
0
200
400
600
800
1000
1200
1400

Defect_4_Images



Problem area: Images with large mask areas

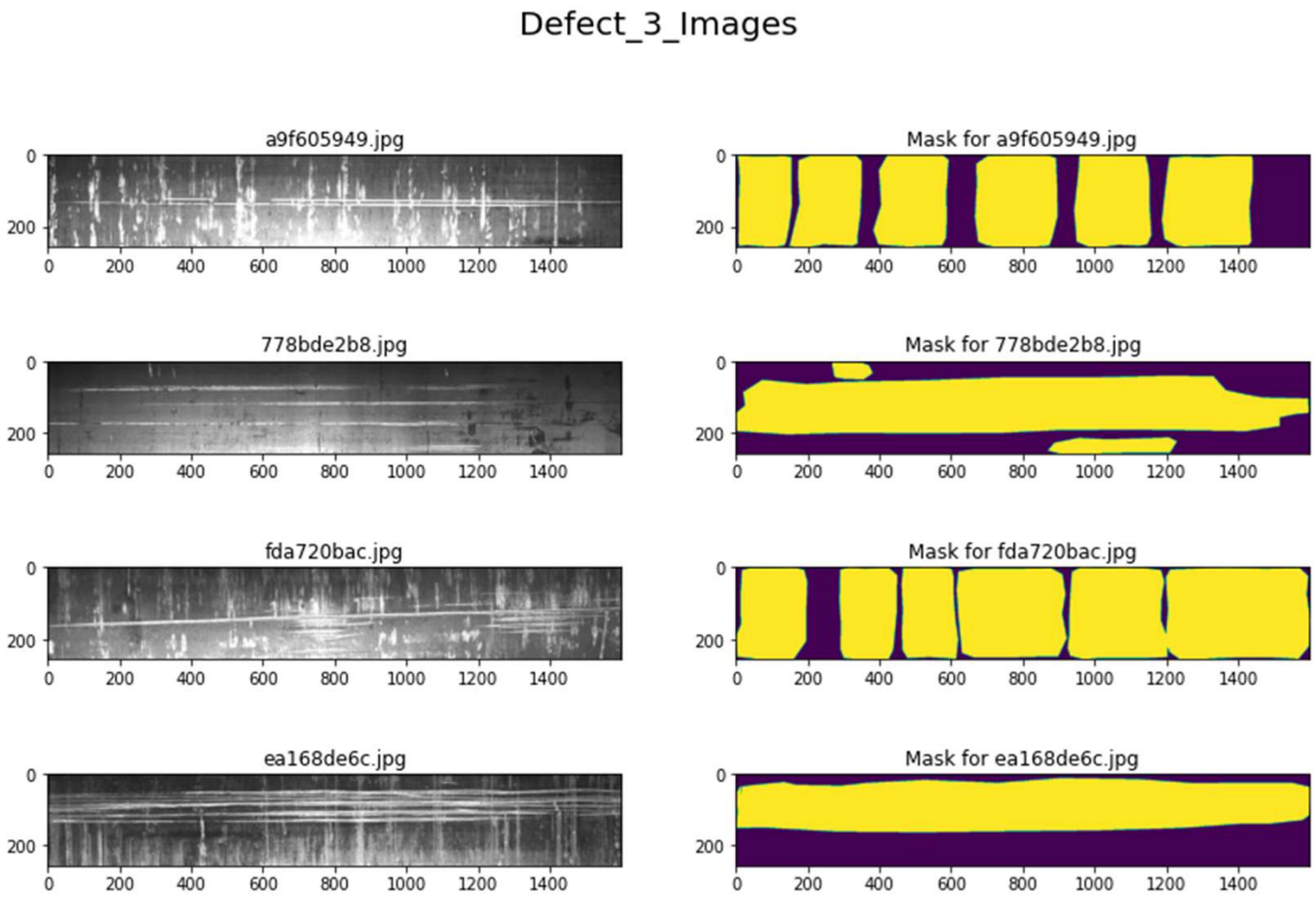


409,600 locations.

rle_Masks > 200,000

Masks covered more than 50% of the image.

Problem area: Images with large mask areas

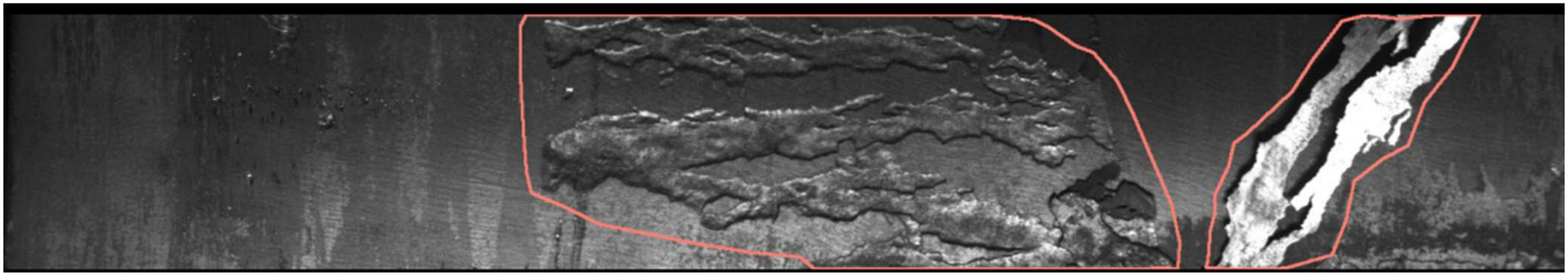


Few samples with large mask areas were originally labeled as ‘Defect_3’.

Visually and by mask_size, they belong to the class ‘Defect_4’.

By industry standards they belong to ‘Defect_3’

Mask to Contours: OpenCV



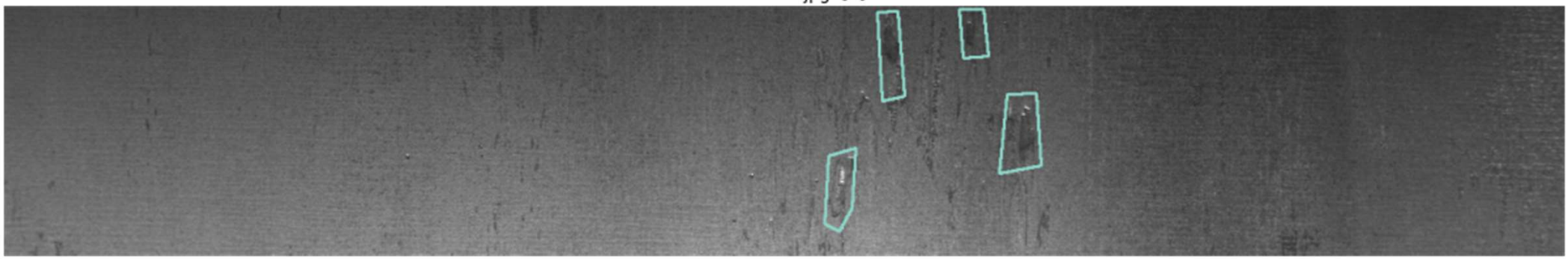
Contours are curves joining all the continuous points; Highlights.

[OpenCV: Contours : Getting Started]

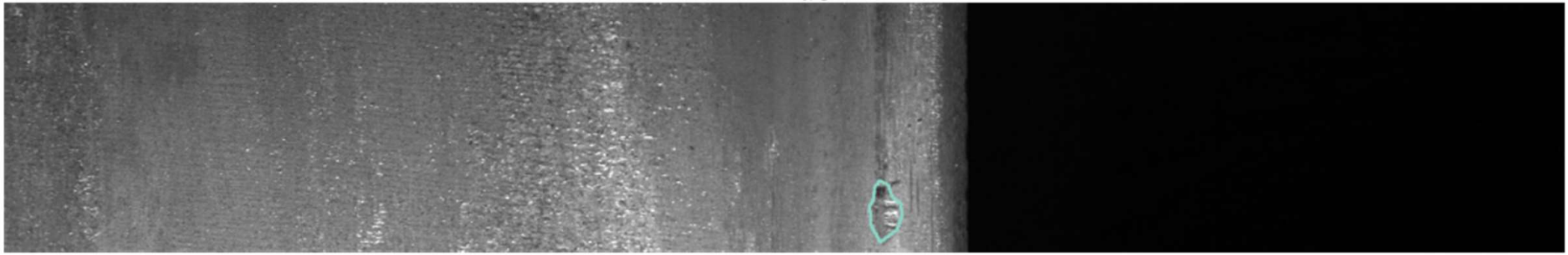
Mask dataframe ⇨ OpenCV contour

Contours for defect_1

e006b532b.jpg: [1]

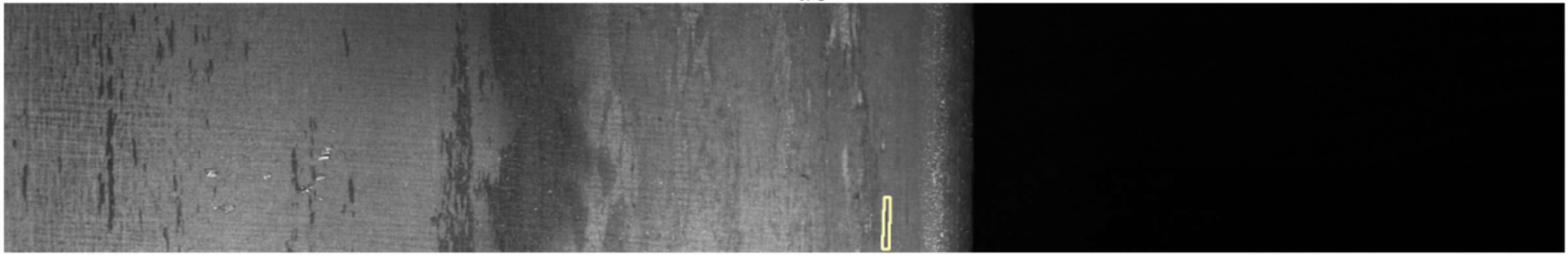


6caa89d4e.jpg: [1]

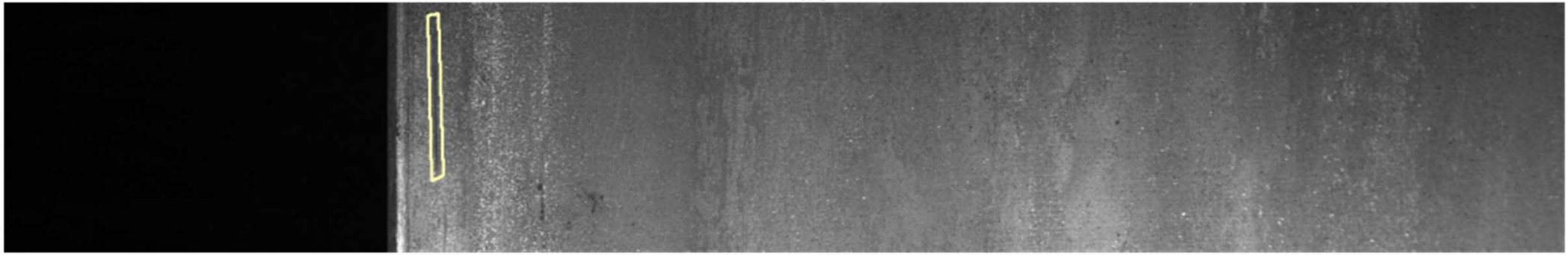


Contours for defect_2

dc58fa33b.jpg: [2]



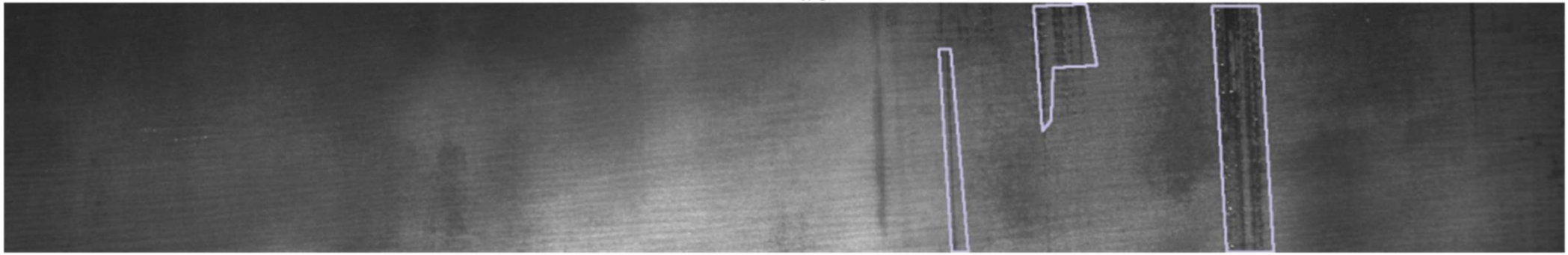
332e21cbc.jpg: [2]



Contours for defect_3

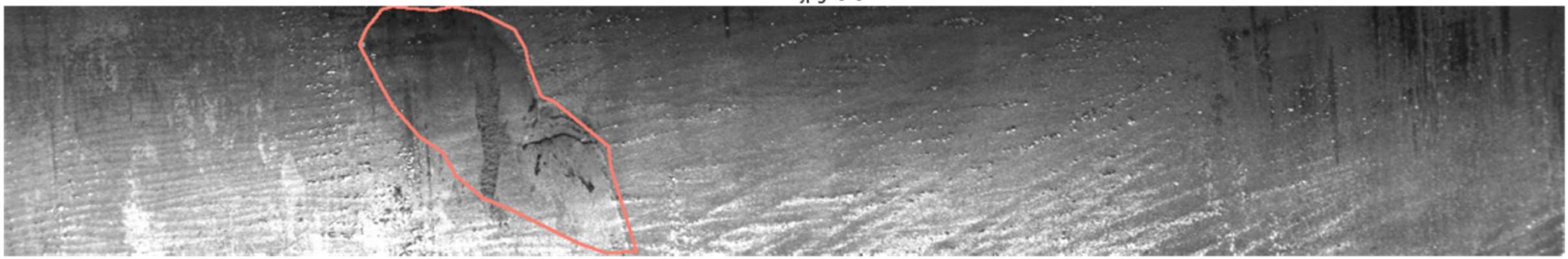
be447691b.jpg: [3]

de79fa864.jpg: [3]

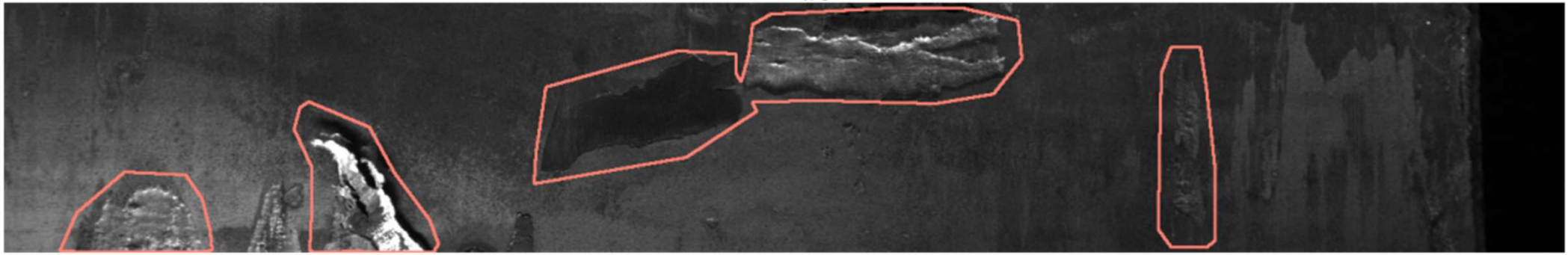


Contours for defect_4

3933a5fcc.jpg: [4]

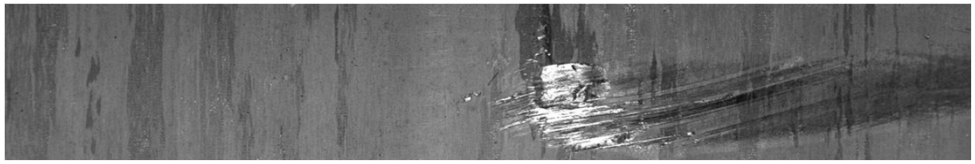


c94629328.jpg: [4]

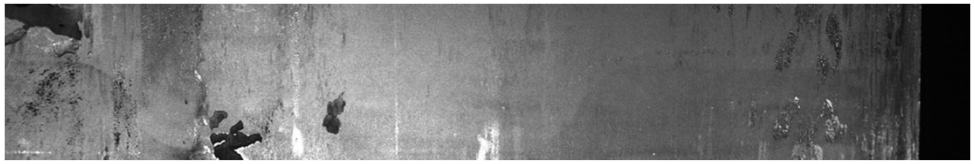


Future Scope - Classification of defects

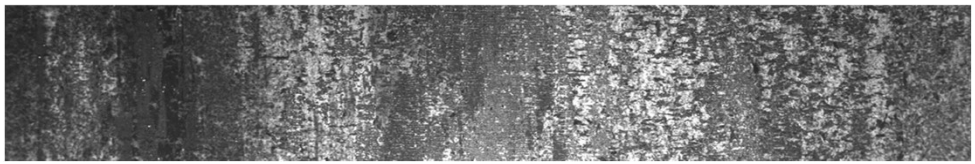
SC - scratches



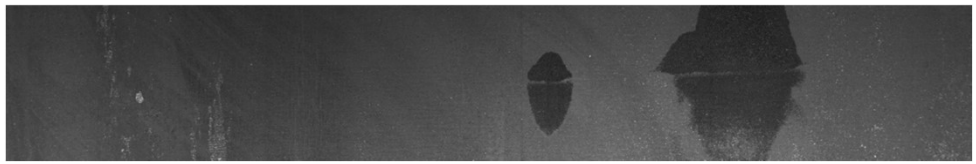
NmIN - Non-metallic inclusions



PS - Pitted Surfaces (caused by salt or Chlorine)



PA - Patches



References

[1] The Sequential model (keras.io)

[2] OpenCV - Invert Mask - GeeksforGeeks

[3] Image Thresholding and Masks with OpenCV - FreedomVC

[4] OpenCV: Contours : Getting Started

[5] Desmos | Scientific Calculator