LARGE SCALE SEMANTIC SEGMENTATION OF VIRTUAL ENVIRONMENTS TO FACILITATE CORROSION MANAGEMENT

R. L. Garcia[1],*P. N. Happ[1], R. Q. Feitosa[1]

[1] Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
rlsg.mec@aluno.puc-rio.br, (patrick, raul)@ele.puc-rio.br

Commission II, WG II/6

KEY WORDS: Semantic Segmentation, Panoramic Imagery, Corrosion, deep learning.

ABSTRACT:

This paper reports the results of a study that aims to develop semi-automatic methods for assessing the degree of corrosion in industrial plant. We evaluated two fully convolutional networks (U-Net and DeepLab v3 +) to segment corroded areas in panoramic images of offshore platforms. The experimental analysis was based on two datasets built for this study. The datasets comprise 9,112 2D images and 3,732 panoramic images. Both FCNs trained on 2D images were tested on 2D images and cubic projections of panoramic images. In addition to pointing out encouraging results, the experiments indicated that most prediction errors concentrated in corrosion defects with a small pixel area.

1. INTRODUCTION

Currently, visual inspection is the primary form to monitor some types of corrosion in industrial facilities. It requires a person to assess damage based on pre-classified visual patterns. Therefore, the procedure carries a high degree of subjectivity, is susceptible to varying knowledge and personal experience biases and is time-consuming. Besides, the produced reports are challenging to visualize and often provides insufficient support for decision-making.

Early works on automatic corrosion segmentation relied on filter engineering to extract texture descriptors upon which corrosion spots could be delineated (Liu et al., 2019), therefore, overcoming some of the limitations mentioned above. Recent approaches such as (Shi et al., 2021), (Papamarkou et al., 2021) and (Fondevik et al., 2020), rely on deep learning techniques to segment corrosion. These works used relatively small datasets comprising less than 900 annotated image samples captured by cameras with a narrow field of views (fov) at a short distance with corrosion spots in focus or close to the centre of the image.

An automatic model must be robust against noise, differences in lighting patterns, and variations in scales and pose to operate in realistic scenarios. Many image samples are necessary to train convolutional networks to adequately represent all the variability of the input images in the operating conditions. This work is concerned about reducing annotator bias and building a model robust to complex scene variations.

In recent years, hardware costs for capturing panoramic images have reduced considerably. Thus, cameras with a large fov are becoming more and more attractive to assess the state of entire industrial facilities. However, these images are highly distorted, and the translation invariance underlying the convolution operation does not hold, which makes the use of conventional deep learning techniques more complex.

This work reports the first results of evaluating two state-of-the-art deep learning architectures, specifically a U-Net (Ronneberger et al., 2015), and a DeepLab v3+ network (Chen et al., 2018), for the segmentation of corrosion spots in industrial facilities from 360° images.

We first built a dataset of annotated 2D images - narrow field of view (fov) - that captures a wide variety of imaging conditions. We also built a second annotated dataset of 360° images for qualitative and quantitative assessment. Six experienced professionals contributed to the data annotation process.

The datasets mentioned above were the basis for a series of experiments that evaluated fully convolutional networks trained on 2D image samples to delineate corrosion spots on 2D and in 360° images of offshore facilities.

The remainder of this paper is organized as follows. The following section describes the methods adopted for corrosion segmentation. Section 3 details the experimental analysis, including the dataset description, the experimental protocol and the performance metrics adopted. The results are presented and discussed in Section 4, and, finally, the conclusions and future directions are summarized in Section 5.

2. METHODS

2.1 U-Net

The U-Net was introduced in (Ronneberger et al., 2015) for biomedical applications right after the first work on fully convolutional network (FCN) was published (Long et al., 2015). The U-Net is an evolution of the basic FCN. It is built up of two paths. The first one, called encoder, reduces the spatial resolution to form a compact representation of the input image, which feeds a second (decoder) path, consisting of a sequence of upsampling operations that recover the input image original resolution. A further distinguishing characteristic of U-Net is the so-called skip connections. They concatenate features produced along the encoder path with the corresponding features calculated through the decoder path. In this way, U-Net recovers fine spatial details that might have gone lost through the successive

* Corresponding author

downsampling operations of the encoder. Figure 1 from (Ronneberger et al., 2015) illustrates the U-Net architectural details.

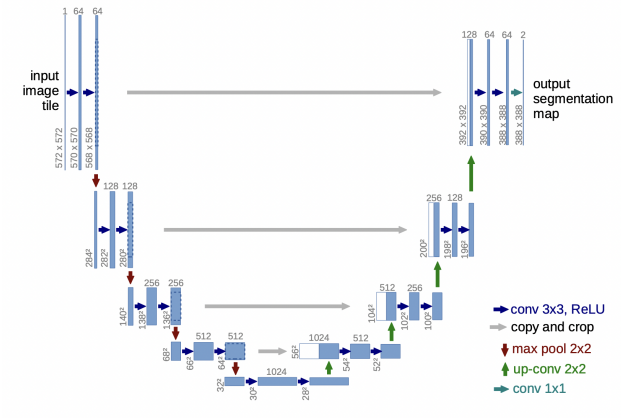

Figure 1. U-Net architecture from (Ronneberger et al., 2015)

Both encoder and decoder paths are composed of blocks with two 3×3 convolutions, each followed by a rectified linear unit (ReLU) activation function. 2×2 max-pooling layers perform the downsampling operations with stride 2. The number of feature maps doubles after each downsampling step. Each block in the decoder path involves a transposed convolution, whose results are concatenated with the corresponding feature maps computed in the encoder path. The encoder and decoder follow the same pattern of two 3×3 convolutions followed by ReLU activation functions.

In this work, we replaced the original U-Net encoder with a ResNet-50 (He et al., 2016) pretrained on the ImageNet dataset (Krizhevsky et al., 2012).

Table 1, shows the ResNet-50 building blocks. The ResNet-50 downsampling operations are carried out first by a 7×7 convolution followed by a 3×3 convolution, both with stride 2. Next, three downsampling operations are carried out by convolutions with stride 2, resulting in a total reduction of spatial resolution by a factor of 32 (same as U-Net). The feature maps brought by skip connections to the decoder path come from encoder layers preceding each downsampling operation.

layer name	output size	50-layer
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times3$
conv3_x	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times4$
conv4_x	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times6$
conv5_x	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times3$
	1×1	average pool, 1000-d fc, softmax
FLOPs		3.8×10^9

Table 1. ResNet50 building blocks (adapted from (He et al., 2016))

2.2 DeepLab v3+

Since its publication, the DeepLab v3+ architecture (Chen et al., 2018) has achieved outstanding performance in many semantic segmentation problems. It uses atrous convolutions to capture spatial context at multiple scales without increasing the number of network learnable parameters, a strategy that has been explored by all DeepLab versions since the first release in (Chen et al., 2014).

The architecture is illustrated in figure 2. Like U-Net, it comprises an encoder and a decoder path. The encoder is a modification of the Xception 65 architecture. The main differences are: it has more layers, replaces regular convolutions with depthwise separable convolutions, replaces max-pooling operations by strided depthwise separable convolutions, and adopts batch normalization and ReLU activation functions just after each depthwise convolution (see figure 3).

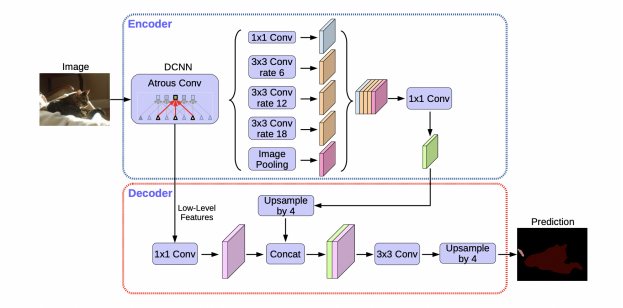

Figure 2. DeepLab v3+ architecture, image credits to (Chen et al., 2018).

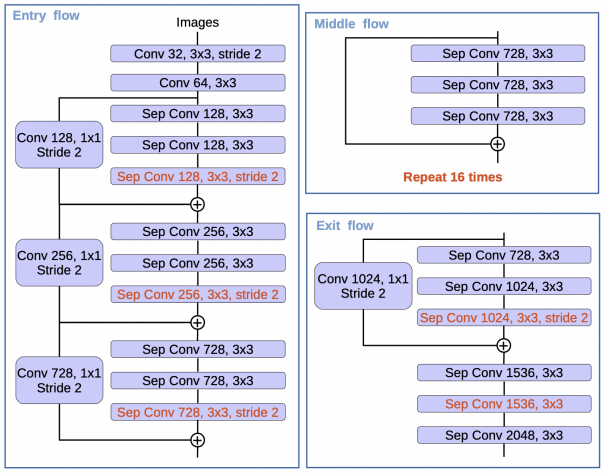

Figure 3. DeepLabV3+: Modified Xception (from (Chen et al., 2018))

A spatial pyramid with parallel atrous convolutions (ASPP module) explores the encoded features to extracts new features at multiple scales. The ASPP module also incorporates an image pooling to capture the global image context. The feature map delivered by the ASPP undergoes a 1×1 convolution, whose outcome is upsampled by a factor of 4 before being concatenated with the result of a 1×1 convolution applied to feature maps computed in the encoder path. A 3×3 convolution followed by an upsampling by 4 completes the model.

2.3 Dealing with panoramic images

Panoramic, also called 360°, images contain RGB data gathered from a scene over a whole spherical field of view. Spherical data can be represented as images using map projections (Snyder, 1987). The commonly used Equirectangular projection unwraps the sphere to a rectangular surface, mapping the meridians to equally spaced vertical lines and the latitude circles to equally spaced horizontal lines. Figure 4 shows an example of an Equirectangular projection.

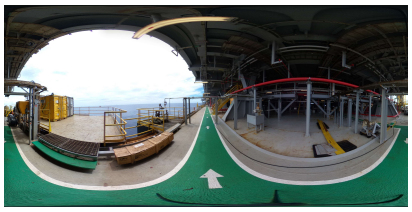

Figure 4. Example of an equirectangular projection

Another way to represent spherical data is the Gnomonic projection, also known as rectilinear projection. This projection maps the sphere to tangent planes. The transformation is described in details in (Snyder, 1987). A simplification of Gnomonic projections is the cubemap projection. Cubemaps are widely used in computer graphics due to their simplicity (Szeliski, 2010)(Snyder, 1987). The method consists of projecting the spherical image into the six faces of a cube that circumscribe the sphere. Figure 5 shows an example of the projections obtained from the image in figure 4 .

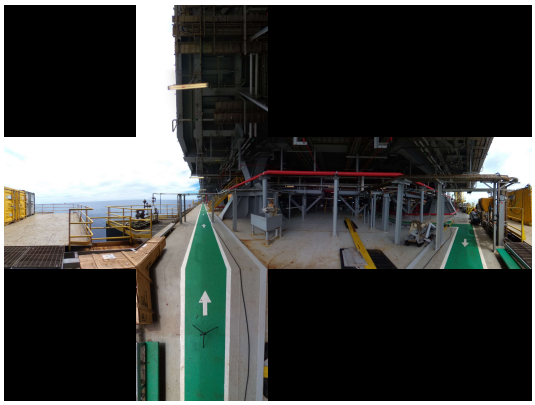

Figure 5. Example of a cubemap projection

In the present study, we used cubemaps to generate 2D rectilinear inputs to FCNs trained on a 2D image dataset to predict pixel-wise class labels.

3. EXPERIMENTAL ANALYSIS

3.1 Dataset

The 2D dataset built for this research comprises 9,112 RGB images of varied sizes. Pixels were labelled as *corroded* or *not corroded*. The images captured in many campaigns represent different poses under various illumination patterns. In total, six annotators, whose main activity is visual corrosion inspection on offshore installations, cooperated in building the dataset. To avoid errors caused by fatigue, a limit of 25 images per day was established per expert.

Annotation of corrosion spots is no simple task because the borders of affected areas are ambiguous, and the subjective perception of the annotator strongly influences the outcome. Figure 6 shows examples with reasonably well-defined corrosion spots in the first and second columns. On the other hand, samples 3, 4 and 5 exemplify a few types of common ambiguities: corrosion stains, marks on the coating provoked by corrosion, but harmless to integrity (3 and 4), left out regions, when the annotator concentrated on the most critical damage (4 and 5).

We further built a second dataset consisting of 3,732 panoramic images for qualitative assessment. A subset of 42 panoramic images was labelled for a quantitative assessment on cubemap projections. The set of labelled 360° images refers to two sites: the newest and least degraded one and the oldest and most degraded one.

3.2 Networks' implementation and training

Classical data augmentation operations, such as horizontal and vertical flips, rotation, shear and translation, were used to enlarge the 2D training set. For training, we used 6,196 images patches of size 513×513 pixels. The validation and test sets had a total of 1,549 and 1,367 images, respectively. We fine-tuned the ResNet-50 pre-trained on ImageNet (Krizhevsky et al., 2012) with a constant learning rate with a batch size of 24.

We adopted for the DeepLab v3+ encoder an output stride equal to 8. The rates of the atrous convolutions in the ASPP module were 12, 24, and 32. We used a model pre-trained on ImageNet, and MS Coco (Lin et al., 2014).

The loss function was the weighted cross-entropy. After having trained on 2D data, we applied both networks to the panoramic images in the following way. Each panoramic image was unfolded into six 2D perpendicular projections, which were given to the tested FCNs. The generated outcome was back-projected to the spherical space and compared with the 360° reference.

3.3 Performance metrics

Let's define *true positives* (TP) as the number of correctly classified *corroded* pixels. Let's further define as *false positives* (FP) the number of pixels wrongly predicted as belonging to the class *corroded*, and as *false negatives* (FN) the number of *corroded* pixels not predicted as such. We define $Precision$ (P) as the proportion of pixels predicted as *corroded* that are actually *corroded*. Formally:

$$P = \frac{TP}{TP + FP} \tag{1}$$

Similarly, $Recall$ (R) is defined as the proportion of *corroded* pixels predicted as such, i.e.,

$$R = \frac{TP}{TP + FN} \tag{2}$$

The f1-score is a popular metric defined as the harmonic mean between $Precision$ and $Recall$. Formally:

$$\text{F1-score} = 2\frac{P \cdot R}{P + R} \tag{3}$$

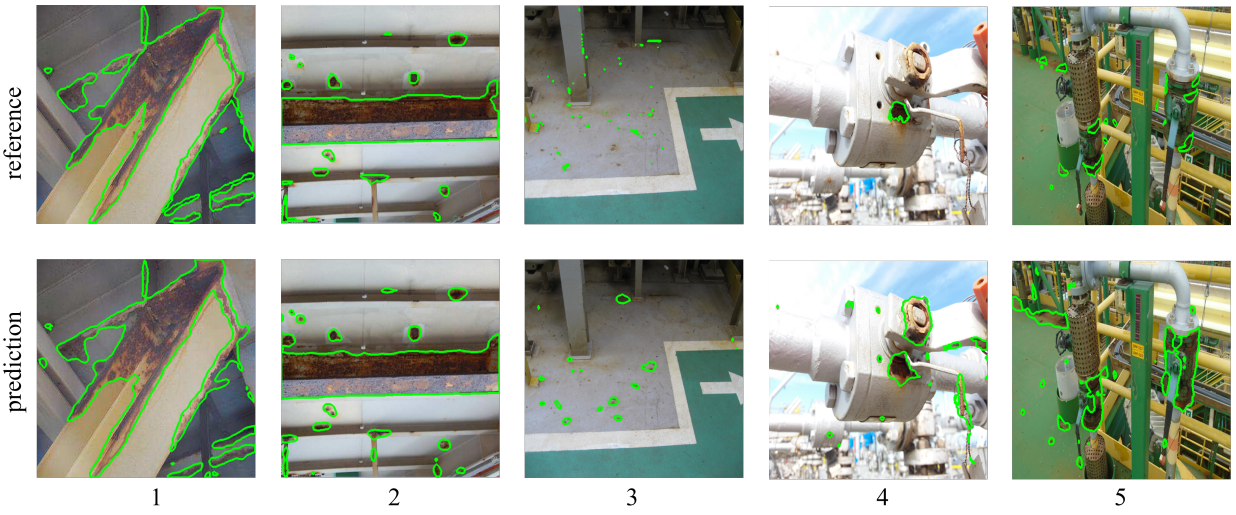

Figure 6. Sample images: reference segmentation (upper row) and example of prediction (lower row)

The average f1-score is given by the arithmetic mean of f1-scores computed for the *corroded* and for the *not corroded* classes.

Another accuracy metric used in our analysis is the *intersection of Union* (IoU). Let's denote with Ref the set of $corroded$ pixels in the reference, and Pre a predicted $corroded$ region. IoU is defined as:

$$\text{IoU} = \frac{|Ref \cap Pre|}{|Ref \cup Pre|} \tag{4}$$

where $|\cdot|$ is the cardinality operator.

IoU can also be computed for $non\ corroded$ regions. The *mean IoU* is given by the average of the IoU values for *corroded* and *corroded* regions.

4. RESULTS

4.1 Evaluation on 2D images

Table 2 shows the accuracy obtained by U-Net and DeepLab v3+ on 2D test images, specifically the f1-score and IoU for the class *corroded* and the mIoU that also considers the *non corroded* class. The DeepLab v3+ model outperformed the U-Net (over 9%) in all the performance metrics. Recall that DeepLab v3+ was pre-trained on ImageNet and Coco while U-Net was pre-trained only on ImageNet, which may have favoured DeepLab v3+.

In the following, we concentrate on the results produced by DeepLab v3+ since it performed significantly better than U-Net, although most conclusions drawn henceforth also apply to the results obtained with U-Net.

Networks	Performance Metrics		
	f1	IoU	mIoU
U-Net (ResNet-50)	53%	39%	59%
DeepLab v3+ (Xception 65)	62%	49%	68%

Table 2. Overall performance on 2D images.

A closer analysis of DeepLab v3+ results reveals that it performed well for large corrosion spots and poorly on images with corrosion circumscribed in small regions in the image. This conclusion is supported by Figure 7 that plots the f1-score vs the percentage of corroded pixels per image. The figure shows a clear trend of the network to perform better on images with large corrosion spots. The other way around, the networks performed poorly on images with a small fraction of corrosion.

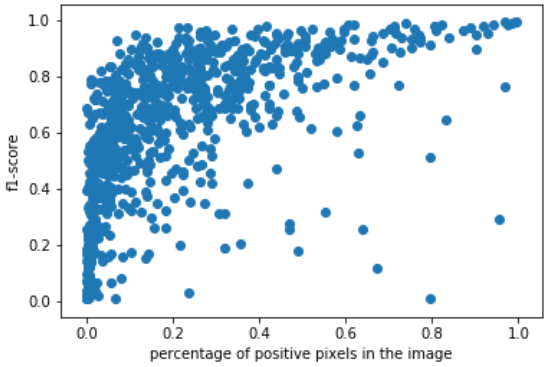

Figure 7. f1-score vs percentage of *corroded* pixels

Figure 8 shows the results from another perspective. It presents the percentage of *corroded* pixels per connected component in the reference image. The horizontal axis represents the average area per corrosion spot, i.e., the percentage of *corroded* pixels divided by the number of connected components on the image label. The vertical axis shows the performance metric, either the f1-score or IoU. The color scale refers to the percentage of positive (*corroded*) samples over the whole image. One infers from Figure 8 that the average area of *corroded* spots more than the image-wise percentage of *corroded* pixels is determinant of networks performance.

Samples 1 and 2 of figure 6 (lower row) show examples of good outcomes produced by DeepLab v3+. They correspond to points 1 and 2 in the plots of figure 8. Even not belonging to the group with a sizeable corroded percentage (less than 30%), these images have large average areas per corrosion area, which places them at the right side of the plots in figure 8.

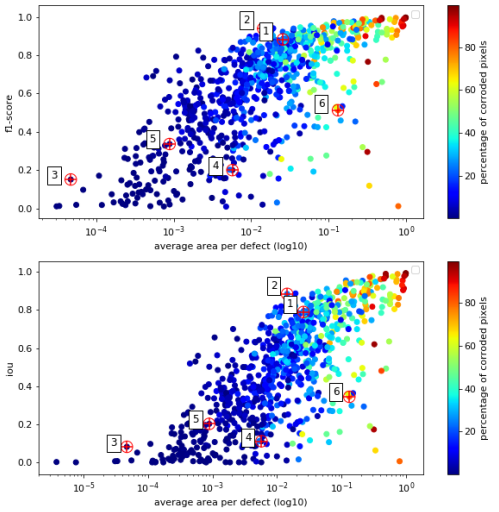

Figure 8. f1-score (above) and IoU (below) versus the percentage of true *corroded* pixels

Points 3, 4 and 5 in the plots of figure 8 correspond to the predictions 3 to 5 shown in figure 6. Looking at these figures, one understands that the poor performance attained on these images was because the regions delineated as *corroded* by the network were considerably more extensive than the actual *corroded* area.

As mentioned in Section 3.1, the datasets contain noisy samples due to the subjectivity of the annotation process. Remarkably, for some test images, the network managed to delineate the *corroded* spots more accurately than the human annotator himself. One example is shown in Figure 9. This example, among others, points to the robustness of the tested FCN against noisy samples.

Figure 9. Left: Input image, Center: Label defined by the annotator. Right: Belief map produced by the network

4.2 Evaluation on 360° images

Networks	Performance Metrics		
	f1	IoU	mIoU
U-Net (ResNet-50)	31%	19.7%	55%
Deeplab v3+ (Xception 65)	41%	28%	59%

Table 3. Overall performance on 360 images.

Table 3 shows the accuracies recorded in our experiments. As before, DeepLab v3+ achieved better performance than U-Net in terms of IoU and f1-score. The difference in performance observed on 360° images compared to the 2D image counterpart was noteworthy. A further investigation revealed that in the dataset of panoramic images, corrosion occurs mainly in small areas. As discussed in section 4.1, the tested FCNs tended to perform poorly under these circumstances, which explains, at least partially, the comparatively inferior performance observed in the experiments on 360 ° images.

Projection	Metrics			
	f1	IoU	mIoU	*%corroded* areas
Front	35%	23%	58%	6%
Right	34%	22%	58%	5%
Rear	38.5%	22%	58%	5%
Left	39.5%	25%	58.8%	6.9%
Top	31.9%	19%	56.7%	4%
Bottom	47.6%	33.6%	57.8%	17%

Table 4. Performance per cube projection

We further observed that the networks performed better at bottom projections (see Table 4). Such projections had on average 17% of *corroded* pixels, which is significantly larger than the percentage on the other projections. Indeed, bottom projections cover steel floors, which usually have larger *corroded* areas by nature compared to other projections that often capture medium to minor defects.

We applied the trained networks to 3,732 panoramic images. A visual inspection of the outcomes showed that the model could detect the most degraded areas successfully. Though we have no reliable reference to compute performance metrics, we took the percentage of predicted *corroded* pixels as a criterion to rank the images according to the level of damage. Figure 10 shows six examples of network predictions. Images 1, 3, 4, and 6 are samples with large degraded areas. Note that the floor accounts for most of the *corroded* area. In image 2 the model misinterpreted the floor as *corroded*. The image in Region 5 is highly ambiguous, although the model marked it as *corroded*.

5. CONCLUSIONS

In this work, we reported the first achievement of ongoing research to develop a semi-automatic procedure to assess the level of corrosion in offshore facilities from panoramic images. We built two datasets comprising 2D and 360°images. Two state-of-the-art fully convolutional neural networks were evaluated for the semantic segmentation of *corroded* spots. The DeepLab v3+ model presented a superior performance consistently when compared with the U-Net.

The tested networks accurately segmented large areas of corrosion but performed comparatively worse as the areas of corrosion decreased. The networks, therefore, performed reasonably well in the type of defects most relevant to the decision on when to initiate major maintenance interventions.

One of the method's expected benefits is the reduction of subjectivity inherent in visual corrosion assessment. Indeed, in some test images, the networks produced better results than those generated by human annotators.

Finally, we observed that networks trained in 2D images behaved similarly when applied to planar projections of panoramic images, though with lower accuracy. A closer analysis of the outcomes indicated that most accuracy degradation was due to the small areas of corroded spots in the panoramic images than the cubemap projection itself.

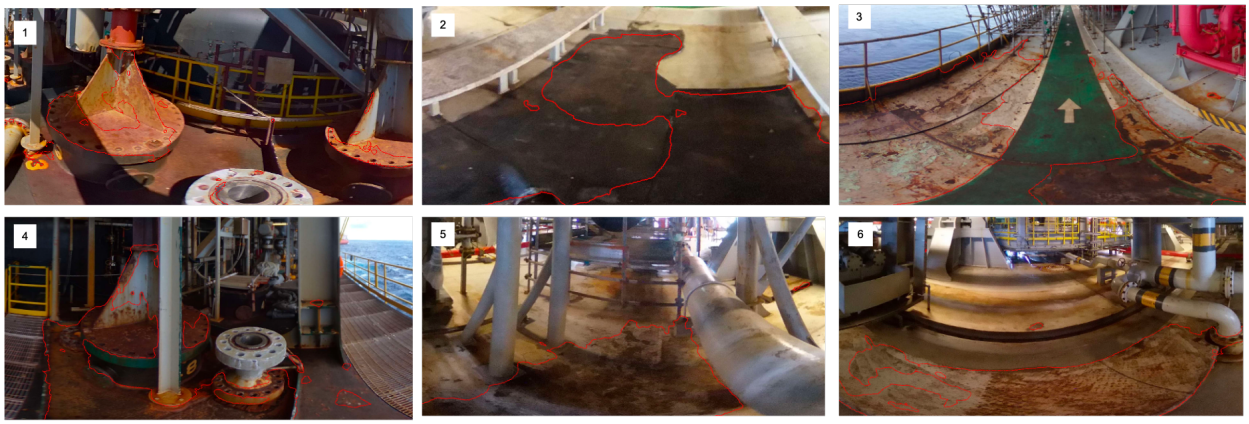

Figure 10. Sample images: examples of reference segmentation (upper row) and prediction (lower row)

The distance to the camera from objects in panoramic scenes of offshore installations varies greatly. The experiments demonstrated that conventional FCNs perform poorly on small *corroded* regions. Even large defects on surfaces away from the cameras will appear as small regions in the images, which will degrade the accuracy. To mitigate this difficulty, in the continuation of this work, it is planned to build a new database that incorporates depth maps.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the valuable support of CNPq, CAPES, FAPERJ and DAAD.

REFERENCES

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV*.

Fondevik, S. K., Stahl, A., Transeth, A. A., Knudsen, O. Ø., 2020. Image segmentation of corrosion damages in industrial inspections. *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 787–792.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097–1105.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European conference on computer vision*, Springer, 740–755.

Liu, L., Chen, J., Fieguth, P., Zhao, G., Chellappa, R., Pietikäinen, M., 2019. From BoW to CNN: Two decades of texture representation for texture classification. *International Journal of Computer Vision*, 127(1), 74–109.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Papamarkou, T., Guy, H., Kroencke, B., Miller, J., Robinette, P., Schultz, D., Hinkle, J., Pullum, L., Schuman, C., Renshaw, J. et al., 2021. Automated detection of corrosion in used nuclear fuel dry storage canisters using residual neural networks. *Nuclear Engineering and Technology*, 53(2), 657–665.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Shi, J., Dang, J., Cui, M., Zuo, R., Shimizu, K., Tsunoda, A., Suzuki, Y., 2021. Improvement of Damage Segmentation Based on Pixel-Level Data Balance Using VGG-Unet. *Applied Sciences*, 11(2), 518.

Snyder, J. P., 1987. *Map projections–A working manual*. 1395, US Government Printing Office.

Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.