AN INNOVATIVE CRACK DETECTION ALGORITHM TO SUPPORT AUTOMATED INSPECTION OF NUCLEAR REACTOR CORES

Efstathios Branikas, Paul Murray, Graeme West
Department of Electronic and Electrical Engineering
University of Strathclyde
efstathios.branikas@strath.ac.uk ; paul.murray@strath.ac.uk ; graeme.west@strath.ac.uk

ABSTRACT

Throughout the nuclear sector, Condition Monitoring and Inspection (CM&I) is used to evaluate, assess and quantify the health and condition of key components and critical infrastructure. Of particular interest is the inspection of nuclear reactors which, depending on reactor design, can be achieved via Remote Visual Inspections (RVI) whereby a camera is deployed within a nuclear reactor to gather images and videos for visualization and analysis. The current manual analysis process is labour-intensive and extremely time-consuming. It is also subjective, prone to human error and difficult to repeat. More recent practices on the detection and analysis of such defects, focus on automating the process, but assign a label on the whole image or a patch of it (image classification). In this paper, we establish a framework for automatically detecting defects from inspection footage to a pixel level, by implementing and modifying a well-known deep neural network. After creating a dataset of images acquired from software that processes and stitches the raw video footage, we train and asses our model. We evaluate the trained model and report the testing accuracy on unseen data. The results are promising and represent an attempt for an automatic pixel-level crack and defect detection system to a field where this family of methods has scarcely been explored: nuclear inspection. This approach provides a low-level localization of cracks, removes bias that is often present with a classification or subjectivity associated with manual approaches and allows repeatability of results.

Key Words: Nuclear crack detection, Deep Learning

1 INTRODUCTION

In the UK, most nuclear power stations are of the Advanced Gas-Cooled Reactor (AGR) design and approaching the end of their expected lifespan. For this reason, inspections are conducted regularly to both ensure the station can prolong its operation and discover, assess, and record damage or defects on various components of the station. In this paper we will present our work on automatically detecting defects in inspection footage obtained from the inner surface of such reactors.

For the family of AGR power plants, the core of the reactor consists of many graphite bricks which are constructed to provide housing for the fuel and control rods while also allowing the movement of CO_2 gas-coolant. The graphite core also acts as a neutron moderator. The inspection of the cylindrical bricks which form fuel channels is commonly called Remote Visual Inspection (RVI) and is conducted using a specialist inspection tool known as NICIE (New In-core Inspection Equipment) which is equipped with a camera among other tools and sensors. The camera is inserted inside the fuel channel and scans its cylindrical inner surface over 6 different overlapping orientations. Any images of cracks or defects detected are stitched into montages that are assessed by experts to further analyze and classify them before allowing the station to return to operation, provided it is safe to do so. This process is usually conducted manually and can be time consuming and costly. Recently, ASIST (Automated Software Image Stitching Tool) software [1], [2] has been developed to automatically create stitched montages of the entire fuel channel by using the raw video footage captured during inspection. The images created are 360 degrees panoramic views of the fuel channel, known as “chanoramas” (channel panoramas). Although ASIST has successfully been deployed

in many hundreds of fuel channel inspections, subsequent processing and quantification of the cracks detected is still conducted manually by human experts, which is time consuming, subjective and difficult to repeat.

Automating the crack detection process poses many challenges; a great deal of inspection data is available, however, only small fraction contains cracks and defects. The conditions of the inspection material vary greatly; different illumination between videos or frames, varying contrast between the defect and its background, cracks of random orientations, thickness, and different scales, all contribute to a very complex and challenging problem. In Figures 1 and 2 some examples of the diversity of the inspection footage can be seen.

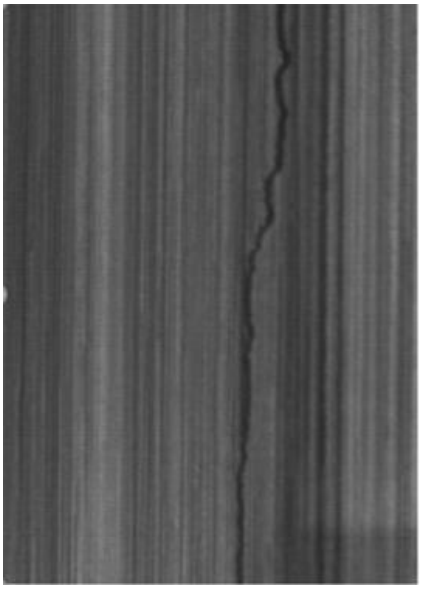

Figure 1. Image of a crack with a rough crack-like texture

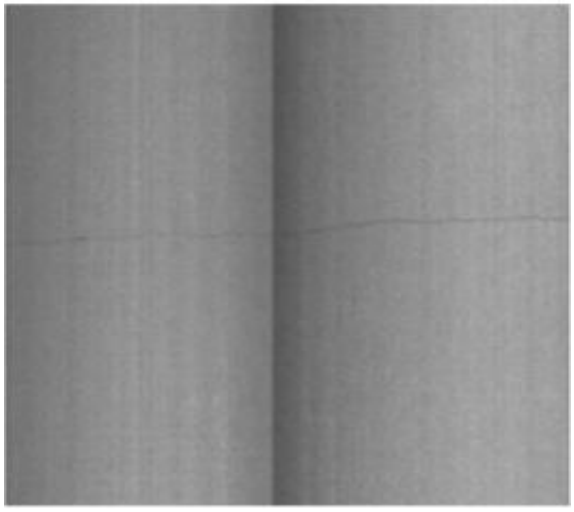

Figure 2. Image of a very faint crack with poor contrast to the background.

The challenges associated with automatically detecting crack from images is not limited to nuclear inspection and a number of related techniques have been presented in the literature. For example, Oliveria et al. [7] extracted histogram statistics and applied dynamic thresholding to identify cracks on road pavement surfaces. Although histogram and thresholding methods provide fast inference, the method suffers from a high rate of false positive predictions. More recently, machine learning and Artificial Neural Networks (ANN) have been widely used for inspection and crack detection tasks providing state-of-the-art results. In [8], Cha et al. modified a region-based Convolutional Neural Network (R-CNN) for detection of defects in buildings and bridges, by providing bounding boxes of the areas of interest. Their work was one of the first to deploy a CNN model for object detection of structural defects. These advances are already being incorporated in the field of structural health inspection and defect prediction in similar applications as our work. In [9], research has been carried out for detection of defects in the nuclear industry where the authors proposed a CNN based framework for predicting cracks from different frames by registering specific patches across different frames and using prediction thresholds. In [10] a similar spatiotemporal registration for video inspection footage from nuclear power stations was conducted utilizing a CNN to classify patches across different frames and follows by a stochastic fusion to increase the accuracy of the patch classification.

These methods classify images or patches of the inspection footage, but do not provide pixel-level localization of the detected defects. They add uncertainty and bias because of this lack of low-level class

prediction. On the contrary, crack segmentation models provide pixel level predictions. For instance, in [11] the authors indirectly performed a segmentation of road pavement surface by classifying the middle pixel of patches and shifting the patches across the whole image. This is computationally prohibitive for nuclear video footage, because of the size of the inspection data. In [12], which inspired our work, the authors deployed U-Net [4], a well-known auto-encoder for detecting cracks on road pavement surfaces. The model was fed with patches and reported very high prediction accuracy on a pixel level. This is the first work to deploy an auto-encoder like U-Net for defect detection, and many similar pieces of research followed. In [13] the authors deployed SegNet [14], an autoencoder similar to U-Net for the segmentation of multiple frames of inspection videos on nuclear stations. In other closely related work, Devereux et al. [3] created a framework that automatically detects and quantifies cracks from chanoramas created from AGR inspection images. However, the authors do not directly provide pixel-level predictions and instead focus on classifying an area of the image and iteratively reduce the area for better localization of the crack.

In this work, we present a new framework for directly performing pixel-level classification (image segmentation) of cracks on inspection data from AGR power stations. We implement and train from scratch U-Net [4], a popular Deep Neural Network belonging to the family of auto-encoders. We feed the model with small patches of images extracted from chanoramas and modify the network to determine the optimal architecture through a validation process. We test our model both in chanorama data, and raw video inspection footage and provide both quantitative and qualitative results to assess the performance of the model.

In summary, the key contributions of this research work can be summarized as creating, to the best of our knowledge, one of the first frameworks for an automatic crack and defect segmentation at a pixel level on nuclear inspection data and one of the first attempts at designing an automatic recommendation system to the human operator of the inspection for detection and sizing of the defects in the raw video footage. The rest of the paper structure can be summarized as: the framework for detecting defects in the nuclear inspection data is defined in Section 2, along with a detailed presentation of the processing of the data; In Section 3 we discuss and analyze the series of experiments conducted, and present the respective results, before we draw our conclusions about them in Section 4.

2 CRACK DETECTION FRAMEWORK

U-Net is a popular Deep Neural Network, consisting of two parts, the encoder and the decoder (thus the name auto-encoder), as can be seen in Fig. 3. The encoder is the contracting path, which is essentially a classifier. It consists of consecutive blocks of operations that compress the initial image input to a tensor of smaller dimensions but drastically more layers, for enriching the feature representation. Each block consists of two sequences of convolutional layers, followed by a ReLU activation and a batch normalization, and finally a down-sampling operation that halves the spatial dimensions but doubles the number of features. For the interested reader, more details on deep learning and convolutional neural networks can be found at [15].

The decoder reverses the down-sampling by stacking blocks that up-sample the information to match the original input. A decoding block consists of transpose convolutions that increase the dimensions of the input by an order of 2 and halves the number feature channels, followed by 2 sequencies of convolutions, activations, and batch normalizations. Each block of the decoder is concatenated with the respective layer of the encoder. This refines the features and ensures that low level information, which is critical for crack segmentation, is preserved. The final output of the decoder is a binary map that corresponds to the predictions for the segmentation.

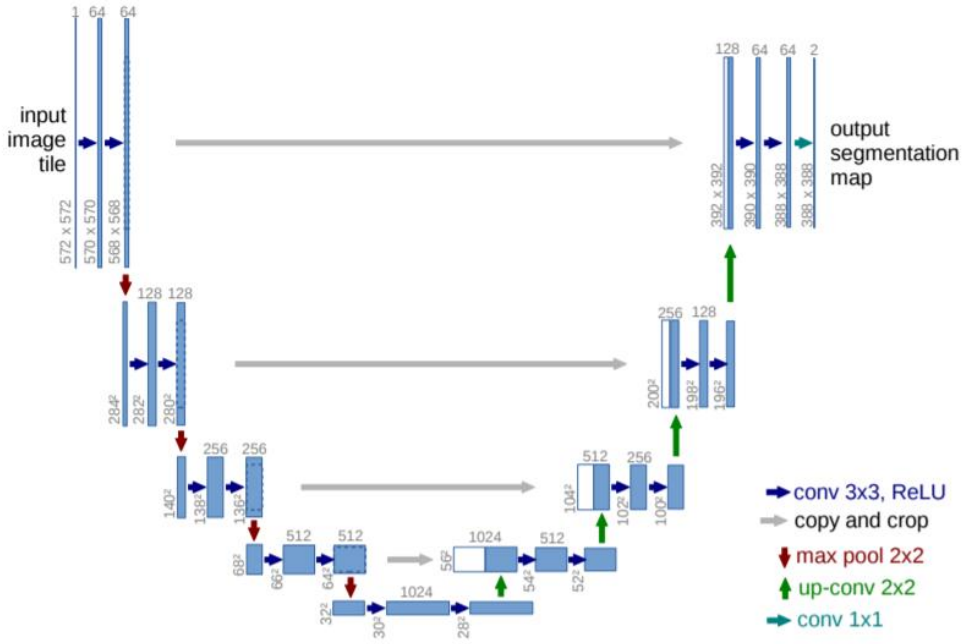

Figure 3. The U-Net architecture, and the respective layers - "U-Net: Convolutional Networks for Biomedical Image Segmentation" [4].

2.1 Data Acquisition

In order to train our model, we construct a dataset of 108 images deriving from chanoramas created from inspection footage. Chanoramas are large grayscale images (150MPixels) so we extract parts of them, corresponding to graphite bricks, that are smaller surfaces, for a more reasonable and flexible handling of the data, and to pinpoint images with cracks. Given this, out of the 108 graphite bricks, 97 of them are used for training the model and the remaining 11 are saved for testing it. We further extract small patches of dimensions of 64 by 64 pixels that overlap, in a shifting window grid manner, by 16 pixels across both dimensions of the image, to acquire a reasonable number of examples for training U-Net.

Inspired by [12], we sample our patches for training to reduce those that do not contain cracks. The reason for this is because the images which do not contain cracks greatly outnumber the cracked patches since cracks, when present, only cover a small surface of the image (5% or even less). If we do not sub-sample the data, the Neural Network will not focus on learning the features corresponding to cracks. In order to tackle this class imbalance, sub-sampling of the patches that do not depict a crack is performed. The subsampling is performed based on the ratio of cracked versus non-cracked patches used during training,. We define a ratio of 50% of training patches to refer to healthy background as optimal, since we observe from experiments that a smaller ratio results in the model having the opposite effect and not learning the different background textures, while a higher reduces the model's prediction accuracy. For the selection, we conduct a series of experiments, corresponding to both random sampling, and based on the entropy value of the grayscale patches. In total, a training set of 78,220 patches is assembled, 15% of which is used for validation and the remaining 85% for training the model.

2.2 Network Training

To train U-Net, we implement the model using Python's Pytorch library. We use the ADAM optimizer for updating the network's parameters through error back propagation and select a batch size of 5 images. At the end of each epoch, we calculate the evaluation metrics on the evaluation set to determine when the model is adequately trained. To further increase the prediction accuracy of the model we explore different modifications of the network and other data augmentation techniques. We substitute the standard convolutional operations with dilated ones, where an extra stride (dilation rate) is inserted to the input in the sense that instead of applying the operation on consecutive pixels, they are sampled based on this stride. This results in a wider field of view for the operation, with no extra computational cost, thus capturing details of different scales, something that is critical for a task like crack detection where the defects commonly appear in different scales and orientations. Importantly, these proposed optimizations are demonstrated to increase the learning capacity of the model as shown in the results presented in Section 3.

With respect to the orientation, we perform an augmentation on the data, by rotating every other positive patch (a positive patch is one containing a crack) by 90 degrees. In this way we provide U-Net with more variety of orientation, something which is quite important, as cracks are root-like, thin-shaped structures which can be present in any orientation.

3 EXPERIMENTAL RESULTS

In order to evaluate the model, we manually create the ground truth for all 109 images using a simple drawing tool that produces a binary mask that is used both for training and testing. We define the abbreviations of the evaluation metrics for the pixel predictions as: true positive: (TP), true negative: (TN), false positive: (FP), false negative: (FN). The metrics are defined as:

$$\text{Dice coefficient} = \frac{2*TP}{(FP+TP)+(TP+FN)} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1 score} = \frac{2*precision*recall}{precision+recall} \quad (4)$$

Based on the framework defined above, we assess the predictive capacity of the model on the testing set, which consists of data the model has not been trained on. For the selection of the optimal parameters, we simply select the epoch with the highest evaluation metrics, from a graph like the one shown in Fig. 4. After selecting the optimal model, we compare the generated segmentation masks with the manually created labor intensive ground truth masks and we report the results in Table 1.

We report the results on different sets of experiments for the 11 "unseen" brick images that were not used in the training of the model (as explained in Section 2.1). The different criteria that distinguish each set are: the sampling of the negative patches: either random or based on the entropy value. The convolutional kernels used in the model: dilated or standard ones. The data augmentation: if positive patches were rotated or not. Under this framework, the experiments consist of all the different combination of these cases, 8 in total.

From the data presented on Table I. various conclusions can be drawn. Rotating the training data for augmentation improves the overall accuracy of the model. The highest recall score is reported for the case of the entropy sampled data, where both the rotation and the dilated kernels were selected. The highest F1 and Precision score was achieved for the random rotation case, using simple kernels. For an application such as crack detection, the recall metric is of the highest importance; the great imbalance between cracks

and healthy background poses the challenge of detecting as many positive (crack) pixels as possible from a small total number, which is reflected by the recall metric. Since the target class is heavily underrepresented and the recall is the rate of the pixels of the target class correctly predicted, it is expected that the recall value will be lower than the precision, but its value is a strong indicator of the model's capabilities. It is also stated that we follow the same approach as in [12] for reporting the model's accuracy; we allow a margin of error of 5 pixels around cracks, where the model predicted a false positive. In this way we do not penalize the model if it predicts slightly thicker masks. These are reported in Table 1 along with the original ones.

Table I. Test results of the different sets of experiments. R: random sampling. RD: random with dilatation. RR: Random with rotation. RRD: Dilation kernels, random sampling, and rotation. Rows with E refer to the respective experiments for the entropy sampling.

	Direct Comparison of cells and ground truth			Comparison with 5-pixel neighborhood compensation		
Method	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
R	0.804	0.752	0.778	0.959	0.805	0.875
RD	0.838	0.761	0.797	0.943	0.781	0.855
RR	**0.877**	0.794	**0.833**	**0.990**	0.813	**0.893**
RRD	0.872	0.787	0.827	0.988	0.807	0.889
E	0.801	0.768	0.784	0.899	0.788	0.840
ED	0.864	0.753	0.805	0.974	0.775	0.863
ER	0.854	0.788	0.819	0.965	0.807	0.879
ERD	0.867	**0.797**	0.831	0.984	**0.817**	**0.893**

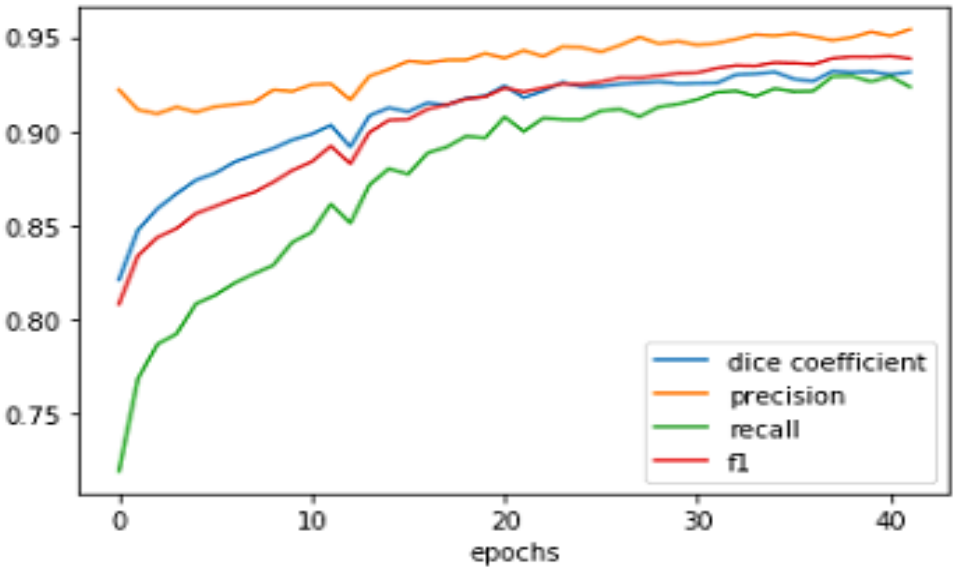

Figure 4. Graphical representation of the evaluation metrics per epoch of training.

In Fig. 5 some examples of the model's predictions are presented. For predicting the class of a pixel, we provide the testing model with shifting patches that overlap so that each pixel has more than one prediction that is mapped to, and a simple majority voting defines the final class to be assigned. From such examples and the metrics reported in Table I. it is reasonable to assume the model performs fairly well. Although there is no ground for comparing with other approaches on this dataset, the evaluations metrics reported during testing in our model are on par with [12] and significantly higher than the results reported in [13] where segmentation of nuclear footage was also conducted. To compare fairly, in the future we intend to implement those approaches on the AGR data, and respectively provide our model with the data used in this related work. However, we further investigate if the results add a fair penalty on the model, as this is the first image segmentation method for such a dataset and both reporting accurate results and setting a fair ground for comparison with future work is critical. Specifically, we investigate if the ground truth penalizes the model because of subjectivity. Since ground truth maps were manually extracted, they are prone to subjectivity and error. In Fig. 6 this is highlighted; it is evident that the manually created ground truth are not completely accurate and contain noise and shapes that do not accurately capture the shape of the crack, therefore resulting in a higher number of false negative predictions, indicated with red in Fig 6.

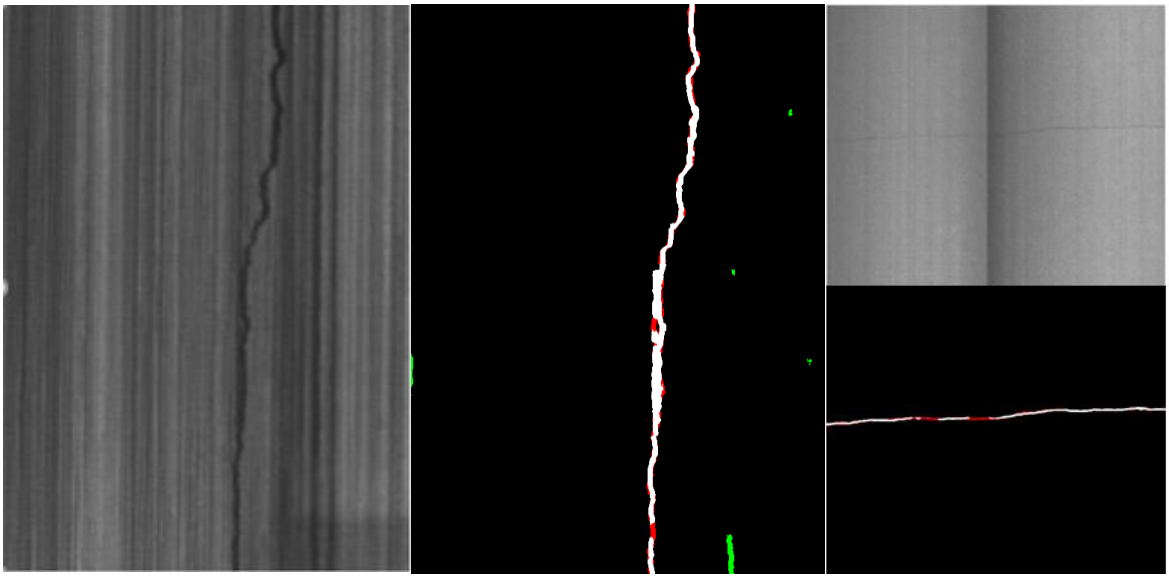

Figure 5. Pixel level prediction of the respective images. White: true positive, red: false negative, green: false positive.

To quantitatively detect the amount of "unfair" penalty the model receives, we choose an approach similar to the relaxation of the false positive pixels mentioned above: we allow a margin around the true positive predictions, where the model's false negatives will not be considered. In this way, any extra thickness or error inserted by the human factor will be discarded and the model will not be rewarded for its actual failures as the margin is negligible. We present the results for a given set of experiments, the case of the entropy sampling with the dilation convolutions and compare it with the relaxed predictions for different margins in Table II.

Based on the contents of Table II and Fig. 6 it is evident that a lot of the error referring to the false negatives can be attributed to the ground truth; the reason for this is that these images are of a quite high resolution, and hundreds of them are available, thus when creating the ground truth, a tradeoff was considered. A pixel-

accurate ground truth is not plausible as it would take a great amount of time to correctly segment all the dataset, therefore the compensation is a time-efficient annotation. In Fig. 6 it is obvious that our assumption reduces this error significantly. With a margin as small as 3 pixels of radius, nearly 10% more of the total pixels labeled as cracks are amended. This finding is significant for image detection applications as the same issue of the human annotation is common across most supervised approaches.

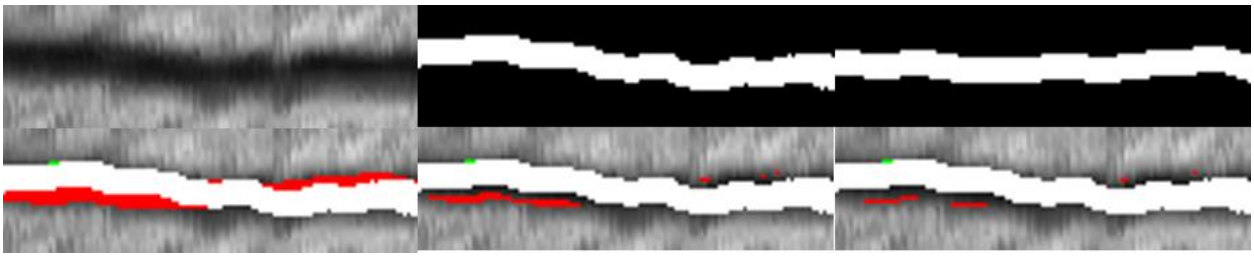

Figure 6. From left to right: upper row depicts a small patch(left), the hand-drawn ground truth(middle) and the model's prediction(right); lower row depicts the original overlayed false negatives(left), the relaxed version with a margin of 3 pixels(middle) and 5 pixels (right) respectively.

Table II. Comparison of the evaluation metrics for different levels of relaxation for false negatives.

Relaxation Radius	Precision	Recall	F1
None	0.974	0.775	0.863
2 Pixels	0.974	0.838	0.900
3 Pixels	0.974	0.870	0.920
4 Pixels	0.974	0.880	0.924
5 Pixels	0.974	0.881	0.925

Furthermore, we assess our model on raw frames, extracted from the original inspection video, the quality of which is lower than the data the model was trained with. The model was deployed on subsets on different inspection videos known to contain cracks - it succeeded in detecting all of them. An example is presented in Fig. 7. Although some noise is inserted to the prediction because of the overlaid metadata, the respective area of the surface of the reactor that corresponds to a crack is correctly marked as positive.

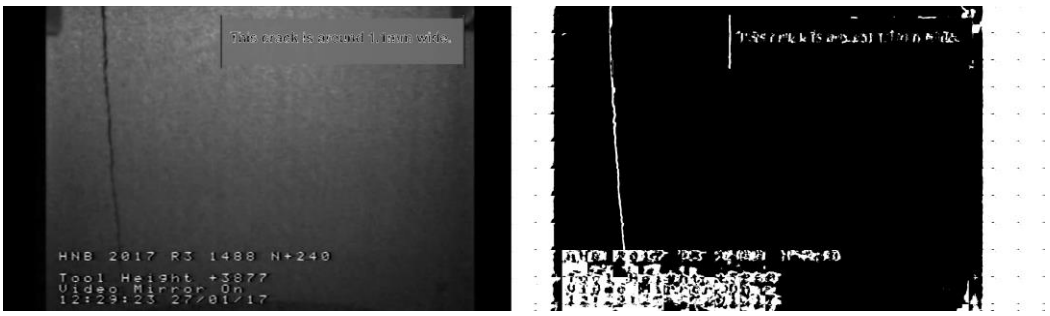

Figure 7. Frame extracted from a real reactor inspection with metadata overlayed (left) and the respective prediction of the model fed with the raw image(right)

We note that we did not apply the model on all the video frames because it is computationally inefficient as their number is of on an order of tens of thousands, but a framework with very low inference time is possible if the model is only fed with a few frames per second and a GPU is utilized. For the purpose of assessing it on long inspection videos we hand-picked the frames where defects were present.

4 CONCLUSIONS

In this research paper we have created a framework for automatically detecting cracks and defects on images and video from inspection of AGR fuel channels by implementing, modifying, and assessing a deep neural network named U-Net. We trained this model based on stitched and processed images known as chanoramas, which are derived from panoramic views of the original inspection video. In parallel, we assess the trained model on the raw video footage to draw a comparison and investigate a possible real-time decision support tool in the inspection process. The advantage of our model is that it provides results of high standards and most importantly, to a pixel level, in contrast to most previous methods and techniques that classify an area of the image, inserting uncertainty about the specific region where the defect is located. Our proposed model provides repeatable results and, for the case of the video application, an inference time that could approach a real time prediction. In the future, we are aiming on making full use of the multi-modal video information, to extract predictions based on multiple frames of information, and refining the ground truth labels in an automated way, thus reducing the labor intensity of the process.

5 ACKNOWLEDGEMENTS

We would like to thank EDF Energy for kindly providing the data that were used in this paper.

6 REFERENCES

1. P. Murray, G. West, S. Marshall and S. McArthur, "Automated in-core image generation from video to aid visual inspection of nuclear power plant cores," *Nuclear Engineering and Design,* **vol. 300**, pp. 57–66, (2016).
2. G. West, P. Murray, S. Marshall and S. McArthur, "Improved visual inspection of advanced gas-cooled reactor fuel channels," *International Journal of Prognostics and Health Management,* **vol. 6** (2015).

3. M. G. Devereux, P. Murray and G. M. West, "A new approach for crack detection and sizing in nuclear reactor cores," *Nuclear Engineering and Design,* **vol. 359**, p. 110464 (2020).
4. O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention* (2015).
5. H. D. Haynes, "Evaluation of check valve monitoring methods," *Nuclear engineering and design,* **vol. 134**, pp. 283–294 (1992).
6. P. Ramuhalli, L. Udpa and S. S. Udpa, "Neural network-based inversion algorithms in magnetic flux leakage nondestructive evaluation," *Journal of applied physics,* **vol. 93**, pp. 8274–8276 (2003).
7. H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *2009 17th European Signal Processing Conference* (2009).
8. Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering,* **vol. 33**, pp. 731–747 (2018).
9. S. J. Schmugge, L. Rice, N. R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe and M. C. Shin, "Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2016).
10. F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Transactions on Industrial Electronics,* **vol. 65**, pp. 4392–4400 (2017).
11. L. Zhang, F. Yang, Y. D. Zhang and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)* (2016).
12. M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy and G. Morison, "A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks," in *2018 26th European Signal Processing Conference (EUSIPCO)* (2018).
13. S. J. Schmugge, L. Rice, J. Lindberg, R. Grizziy, C. Joffey and M. C. Shin, "Crack segmentation by leveraging multiple frames of varying illumination," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2017).
14. V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence,* **vol. 39**, pp. 2481–2495 (2017).
15. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai and others, "Recent advances in convolutional neural networks," *Pattern Recognition,* **vol. 77**, pp. 354–377 (2018).