Small Defect Detection Using Convolutional Neural Network Features and Random Forests

Xinghui Dong, Chris J. Taylor, and Tim F. Cootes

Centre for Imaging Sciences, The University of Manchester,
Manchester, M13 9PT, UK
{xinghui.dong, chris.taylor, timothy.f.cootes}@manchester.ac.uk

Abstract. We address the problem of identifying small abnormalities in an imaged region, important in applications such as industrial inspection. The goal is to label the pixels corresponding to a defect with a minimum of false positives. A common approach is to run a sliding-window classifier over the image. Recent Fully Convolutional Networks (FCNs), such as U-Net, can be trained to identify pixels corresponding to abnormalities given a suitable training set. However in many application domains it is hard to collect large numbers of defect examples (by their nature they are rare). Although U-Net can work in this scenario, we show that better results can be obtained by replacing the final softmax layer of the network with a Random Forest (RF) using features sampled from the earlier network layers. We also demonstrate that rather than just thresholding the resulting probability image to identify defects it is better to compute Maximally Stable Extremal Regions (MSERs). We apply the approach to the challenging problem of identifying defects in radiographs of aerospace welds.

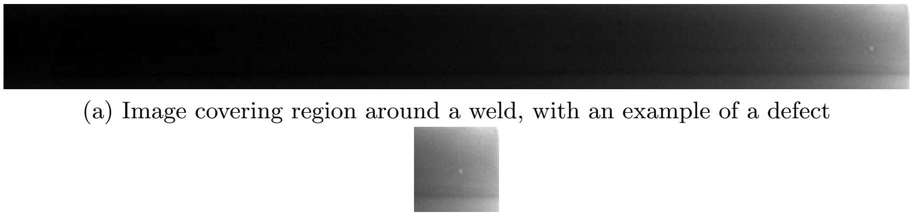
Keywords: Defect detection, non-destructive evaluation, CNN, local features, random forests

1 Introduction

Inspection tasks, where one is looking for small defects in large regions, can be challenging because (1) any useful system must have a very low false positive rate and (2) since defects are rare it is not always easy to obtain large numbers of examples to train a classifier or a region-based object detector. Given their success in other areas [16], it is natural to try Fully Convolutional Networks such as U-Net [21] to label each pixel in a region to indicate those that belong to abnormalities of interest. In this paper we show that when only small numbers of training examples are available, such techniques perform poorly. However, we go on to show that the features that are learnt in the different U-Net layers are useful for discriminating good from bad, and that a Random Forest, trained to classify each pixel based on such features, achieves a much better result than the usual final softmax layer in the U-Net.

The output of such an approach is a label image, giving the probability that each pixel belongs to a defect. A natural approach to identifying defects using such an image is to apply a threshold, then select the connected regions passing this threshold. This approach is prone to false positives. We show that better results can be obtained by computing Maximally Stable Extremal Regions (MSERs) [18] in the probability image, then identifying which are likely to be true defects by examining the distribution of probabilities across the region.

We demonstrate the system on the task of identifying defects in radiographs of welds in aerospace components a safety critical application where currently visual inspection by a human expert is the norm. Figure 1 shows an example of a region extracted along a typical weld [7], and an example of a defect that should be detected. Note that this is a particularly obvious defect many are much more subtle.



(a) Image covering region around a weld, with an example of a defect

(b) Patch

Fig. 1. Examples of a radiograph along a weld [7].

To inspect large regions, we first normalise them and then break them down into patches. Each patch is fed to a U-Net [21]. The U-Net is trained only on patches sampled around the defects in the training set to avoid the system being overwhelmed with normal (non-defect) pixels which are in the vast majority.

The contributions of the work are (1) we describe a system for identifying small defect regions in large images; (2) we show that using a Random Forest at the end of the process significantly improves on the performance of U-Net in this domain; and (3) false positives can be reduced by using MSER to identify candidate regions, rather than a simple thresholding.

The rest of this paper is organised as follows. In the next section, we review some existing work. Our methodology is introduced in Section 3. The experiments are re-ported in Section 4. Our conclusions are drawn in Section 5.

2 Related Work

2.1 Defect Detection Using Images

Non-destructive evaluation (NDE) of manufactured components [7], [1], [14] is critical in many industries such as aerospace. Even small defects can significantly reduce the life of a component, potentially leading to failure and accidents. Human inspection by trained experts is still the norm in the industry, but can be inconsistent, subjective, tedious and time-consuming [14]. There is a need to develop of automatic inspection systems to reduce the workload of inspectors and provide more consistent, objective and efficient decisions.

Kehoe and Parker [11] applied a thresholding method to the statistics of pixels for defect detection. Wang and Liao [25] detected defect candidates using the background subtraction and histogram thresholding approaches. The two-stage method that Wang et al. [26] introduced for line defect detection includes defect detection and defect refinement. Yazid et al. [27] used the inverse surface thresholding technique for defect detection. Using grayscale arranging pairs (GAP) features, Zhao et al. [29] first constructed a model based on the pixel pairs which own the stable intensity relationship. Defect detection was performed by thresholding the difference in the intensity-difference signs between the novel image and the model. A series of basic image processing operations were used by Boaretto and Centeno [1]. Recently, Dong et al. [7] developed a weld defect detection method using Haar-like features [24] and Random Forests [2]. Although this method produced promising results, some defects were still missed.

2.2 CNN-Based Image Segmentation

The above approaches were normally implemented using either basic image processing methods, or traditional features and classifiers. In contrast, Convolutional Neural Network (CNN) techniques [13] have shown state-of-the-art performance in many computer vision tasks. Training a CNN requires a large number of labelled examples. Alternatively, the fully-connected (FC) layer of a pre-trained CNN can be used as a generic feature extractor [19]. Ren et al. [20] used the FC features extracted from a set of image patches to train a classifier. The classifier can be used to predict as to whether or not a patch contains defects. However, extraction of FC features from patches is time-consuming. This issue limits the practicability of the method in the real industrial inspection. Using a large image patch set, Chen and Jahanshahi [3] trained a CNN classifier from scratch for detecting cracks. Naive Bayes decision making was used to remove false positives. However, where only few images are available for training, the system will not work so well.

Using a sliding-window CNN classifier, Ciresan et al. [5] segmented the neuronal membranes contained in electron microscopy images. Girshick et al. [8] applied a region classifier to semantic segmentation. The FC features extracted from a set of regions that were produced by region proposal were first fed into a regressor. Then, the regressor was used to predict the quality score of each region in terms of a class. Hariharan et al. [10] proposed the hypercolumn features that were extracted at multiple CNN layers in order to exploit both the localisation and semantic data. Instead of using a pre-trained CNN as the feature extractor, Long et al. [16] adapted it into a Fully Convolutional Network (FCN) and transferred its representations to semantic segmentation by fine-tuning this CNN using a large domain-specific dataset.

However, such a large dataset may be unavailable in some fields, e.g. medical image processing. To address this problem, Ronneberger et al. [21] developed a novel CNN architecture, U-Net. It comprises a contracting path encoding the context information and a symmetric expanding path that captures the localisation data (see Figure 2). Using even a small dataset, U-Net can be trained end-to-end from scratch and outperformed the method that Ciresan et al. [5] proposed. For segmentation of neuronal cells in confocal microscopy images, it is challenging to label the pixel-wise ground-truth. When only a relatively small set of instance-wise ground-truth images are available, the direct training of FCNs becomes impractical. Chen et al. [4] proposed a method to train pixel-wise Complete Bipartite Networks (CB-Net) using these data. Zhang et al. [28] applied a Deep Adversarial Network (DAN) model to biomedical image segmentation in which the annotated or unannotated images were available.

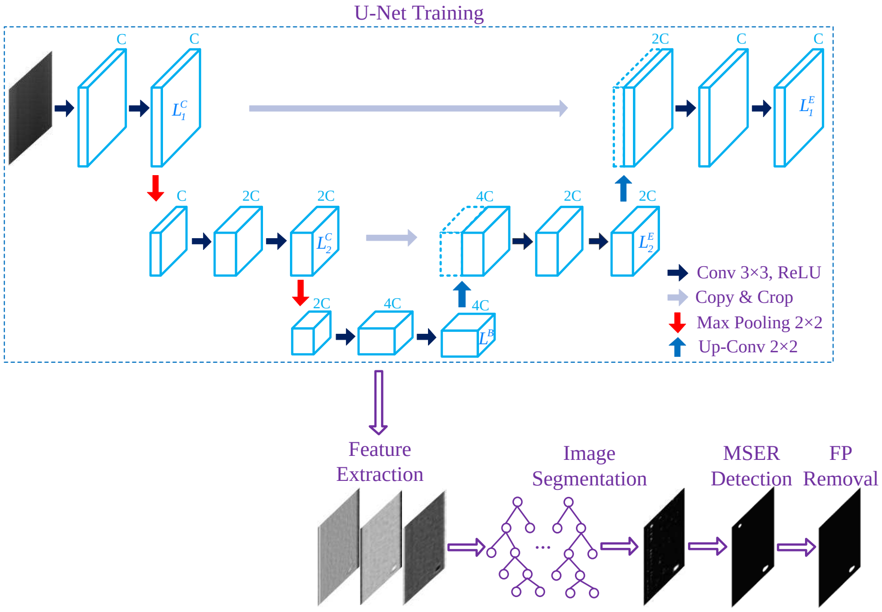


Fig. 2. The flowchart of the proposed small defect detection approach.

3 Methodology

We assume that we have a training set containing example images each containing typical defects, together with binary label images indicating the pixels belonging to each defect. Since defects are small and rare, the defect pixels are a tiny minority of all pixels, making training difficult. To train a U-Net [21], each image is first augmented with rotations, reflections and distortions, and is then normalised by applying a linear transformation to zero the mean and arrange for the variance to be unity. Patches are further extracted around each defect. We train a U-Net to predict the binary pixel labels using this set. We then train a Random Forest to use features extracted at each pixel position in the U-Net to predict the pixel label essentially replacing the final softmax layer of U-Net with a Random Forest classifier. Random Forests are known to generalise well from small numbers of samples [6], [15], so are suitable for this application.

Figure 2 shows the flowchart of the proposed approach to process a new image. The output of the U-Net + RF is a probability image. We compute Maximally Stable Extremal Regions (MSERs) in this probability image, and analyse each to discard false positives. In the following we describe the steps of the training and processing in detail, and show the results of experiments evaluating the performance of the different parts of the system.

3.1 U-Net Training

Data Augmentation Data augmentation has been shown to improve the performance of deep learning methods [23]. We apply a pipeline of augmentation operations to the images and associated labels. This includes shearing, skewing, flipping and elastic distortion operations in turn. The occurrence probabilities for the four operations are 0.5, 0.5, 0.75 and 1.0 respectively.

Normalisation and Patch Sampling We enhance the contrast of each image, and improve invariance to illumination effects, by applying z-normalisation [9] to each image. At each pixel we apply a trans-formation

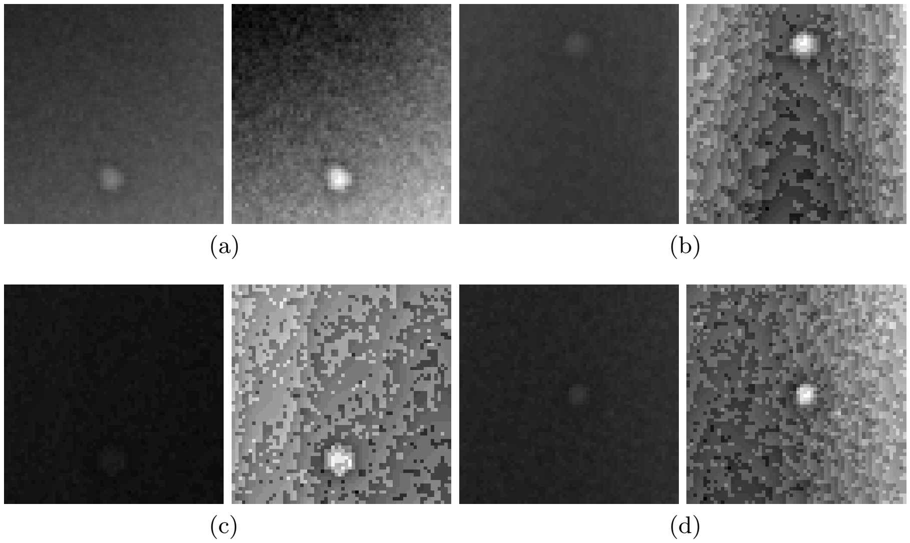
$$p^{'} = \frac{p - \mu}{\sigma}, \tag{1}$$

where μ and σ are the mean and standard deviation computed in the $W \times W$ region centred at the pixel. We sample a $P \times P$ patch around each defect from each image. The patch is taken at a randomly displaced position so that the position of the defect is not centrally biased. Figure 3 shows examples of the original and normalised patches.

Model Training We use the Adam [12] approach for parameter optimisation in U-Net [21], training on the pre-processed patches.

3.2 Feature Extraction Using the Pre-trained U-Net

Using the pre-trained U-Net [21], one can extract features from an image at different convolutional layers. We use the second convolutional layer at each level on one path for feature extraction (see Figure 2 for more details). Given the U-Net contains L levels, the second convolutional layer at the bottom level (i.e. the L-th level) is denoted as L^B. In terms of the contracting (left) and expanding (right) paths at the j-th ($j \in \{1, \cdots, L-1\}$) level, we denote the



(a) (b) (c) (d)

Fig. 3. Examples of z-normalisation ($W = 71$). (Left: Original, Right: Normalised). For display purposes, the image patches have been enlarged.

second convolutional layer as: L_j^C and L_j^E respectively. For an $M \times N$ image, a series of $M^{'} \times N^{'} \times C$ (C is the number of filters) feature maps can be computed at one of the L^B, L_j^C and L_j^E layers. Each feature map is individually resized to the size of $M \times N$ using bilinear interpolation. In this context, each pixel is represented by C features. Motivated by the use of hypercolumn features [10], we also extracted features at multiple layers and concatenate these along the feature channel axis. The feature vectors at each pixel are independently L_2 normalised.

3.3 Image Segmentation Using a Random Forest

We apply the U-Net [21] to the (un-augmented) training set and obtain feature vectors for each pixel. We create a training set of feature vectors by extracting the vectors associated with every positive (defect) pixel and an equal number of (randomly sampled) negative pixels. These samples are used to train a random forest to estimate the pixel label based from the feature vector.

Given a new image, an output probability image is created by applying U-Net, then using the random forest to estimate the defect probability at every pixel.

3.4 Defect Candidate Detection Using the MSER Detector

We use the Maximally Stable Extremal Region (MSER) detector [18] to identify defect candidates from each probability image. The image is first linearly stretched to the range [0, 255]. Compared with applying a threshold to the image, this method exploits the higher level characteristics encoded in connected components. After MSER detection is complete, a set of defect candidates are obtained for each test image.

3.5 False Positive Removal

To evaluate whether or not a region (identified by MSER) is a true positive we calculate the number of pixels, n_p, which have a probability above a threshold value, t. We remove any region in which fewer than $k\%$ of the pixels pass the threshold.

4 Experiments

In this section, we first introduce the experimental setup and performance measures used in our experiments. Then, we report the results derived in the image segmentation and defect detection experiments.

4.1 Setup

We used a set of 43 X-ray images and associated label images collected by Dong et al. [7] from an aerospace manufacturing company. Each image had at least one defect. Defects are rare in aerospace components, making it difficult to obtain a large dataset. We used a 10-fold cross validation scheme for all experiments reported. For each split, data augmentation leads to 10 extra images for each training image. 64×64 patches sampled from the original and augmented images were used for training a single U-Net [21]; while only the original images were used for training an individual random forest classifier [2]. When testing, only the original images were used for both U-Net and random forests. All images (including those used for sampling patches) were processed using z-normalisation with $W = 71$.

U-Net [21] was trained using different levels and numbers of filters (batch size = 8, 100 epochs and 32 iterations per epoch). To train the random forest, feature vectors were extracted from one or more U-Net layers and concatenated. All images or patches were padded with zeros in order to generate the probability map with the same size as the input. Given an F dimensional feature vector, a subset of $\lceil \sqrt{F} \rceil$ features were randomly selected using the random forest classifier. The Gini impurity measure [2] was used to choose the feature and threshold at each node. The minimum number of samples at each terminal node was set to 0.01% of the training set size.

4.2 Performance Measures

We measure the per-pixel classification performance using the Receiver Operating Characteristic (ROC) curve, plotting the true positive against false positive fraction for each of a choice of probability threshold values, t. In order to derive a single quantitative measure, we computed the Area under the Curve (AUC) from a ROC curve. Since defect detection focuses on identifying defects rather than single pixels we use an object level measure, i.e. Free Response Operating Characteristic (FROC), plotting the proportion of defects detected against the number of false positives per image. We consider a defect detected if more than half of its area is included in a positive detection region. We used a threshold of $k = 59$ (chosen empirically using pilot experiments) when discarding false positive MSERs.

4.3 Image Segmentation

We first tested the impact of different experimental parameters on image segmentation. Then, we compared the proposed feature set with other image feature sets.

Impact of the Number of Levels We first trained U-Net [21] using between 1 and 4 levels. Figure 4 shows the ROC curves and AUC values derived using L ($L \in \{1, \cdots, 4\}$) levels U-Net alone. The best result for this data (AUC: 0.9825) was obtained using three levels.

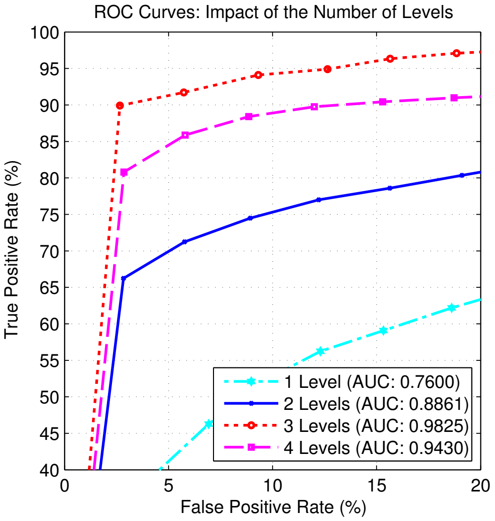


Fig. 4. ROC curves plotted using the results of different numbers of U-Net [21] levels for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

Impact of the Number of Filters We trained 3-level U-Nets using C ($C \in \{32, 64, 128\}$) filters see Figure 5. The highest AUC value: 0.9825 was produced using 64 filters.

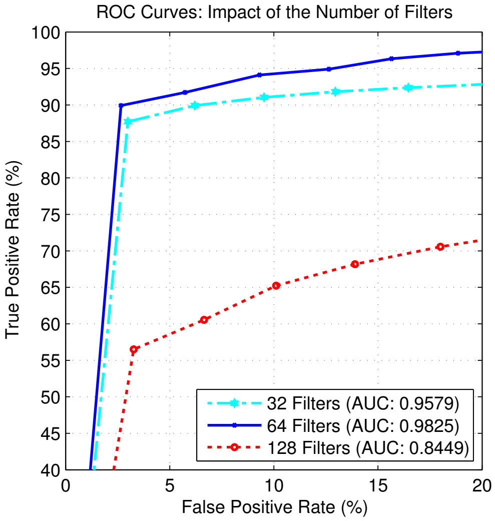


Fig. 5. ROC curves plotted using the results of the 3-level U-Net [21] with different numbers of filters for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

Impact of Z-Normalisation We trained the U-Net with and without z-normalisation [9] on the input patches. As shown in Figure 6, the use of z-normalisation greatly boosted the performance, increasing the AUC from 0.88 to 0.98.

Softmax vs. Random Forests Since the softmax layer is immediately placed behind the convolutional layer: L_1^E in U-Net, we examined the performance of the random forest classifier [2] along with the features extracted at this layer. We used 25 and 200 trees for the classifier individually. The results are compared with that generated by the softmax layer in Figure 7. Although softmax yielded the slightly higher AUC value than those obtained using random forests, the latter produced higher true positive rates when false positive rates were below 11%.

Impact of Different Layers of the Pre-trained U-Net Given three levels were used for U-Net we tested features from five different convolutional layers,

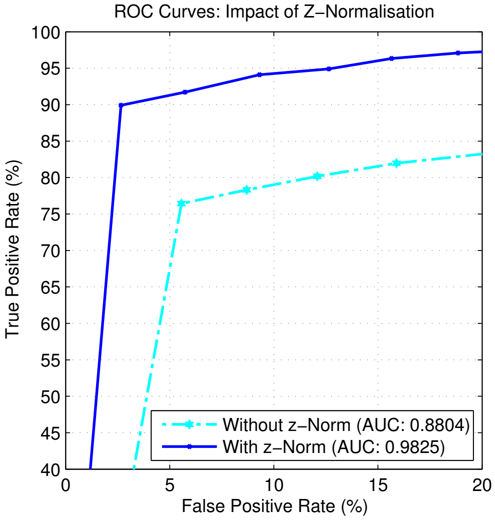


Fig. 6. ROC curves plotted using the results of the end-to-end trained U-Nets [21] with and without z-normalisation for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

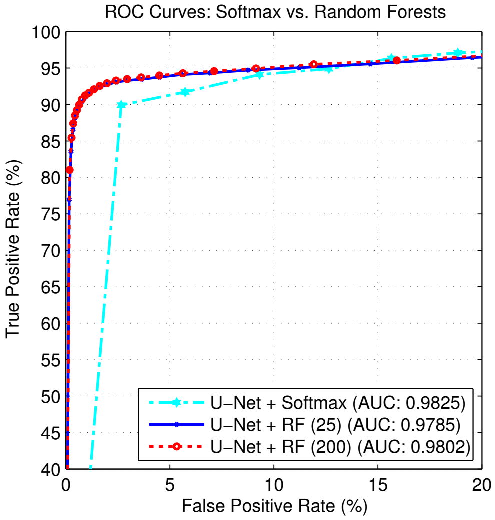


Fig. 7. ROC curves plotted using the results of the end-to-end trained U-Net [21] and the features extracted using this U-Net and random forests for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

L_1^C, L_2^C, L^B, L_2^E and L_1^E (see Figure 2). For each layer, features were extracted and used to train a single random forest classifier. The results are displayed in Figure 8. Features from the bottom convolutional layer, L^B, produced the best results.

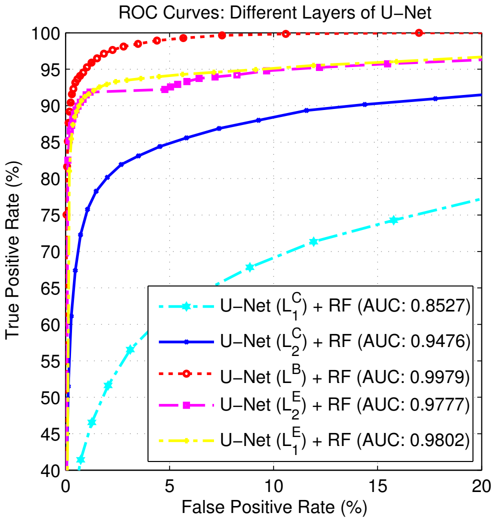


Fig. 8. ROC curves plotted using the results of the different layers of a pre-trained 3-level U-Net [21] for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

Single-Layer vs. Multi-layer of the Pre-trained U-Net Inspired by the application of hypercolumn features [10], we extracted features at multiple layers of U-Net. Table 1 reports the AUC values obtained using the convolutional layer: L^B and nine different combinations of U-Net layers. As can be seen, the single layer L^B produced superior results to those generated by its multi-layer counterparts.

Comparison with Existing Feature Sets Finally, we compared the results obtained using the features extracted at the bottom layer L^B of U-Net [21] with those derived using two existing feature sets, (1) SIFT features [17] and (2) features extracted at the last convolutional layer of a pre-trained VGG-VD-16 [22] model. As presented in Figure 9, the U-Net features outperformed both the SIFT and VGG-VD-16 features with large margins.

Table 1. The AUC values obtained using an RF on features from various layers of U-Net [21]

Layers	L^B	$L_1^E + L_2^E$	$L^B + L_1^E$	$L_1^C + L^B + L_1^E$	$L_2^C + L^B + L_2^E$	$L_1^C + L_2^C + L^B$	$L^B + L_2^E$	$L_1^C + L^B$	$L^B + L_2^E + L_1^E$	$L_1^C + L_2^C + L^B + L_2^E + L_1^E$
AUC	**0.998**	0.973	0.996	0.997	0.995	0.997	0.995	0.997	0.994	0.994

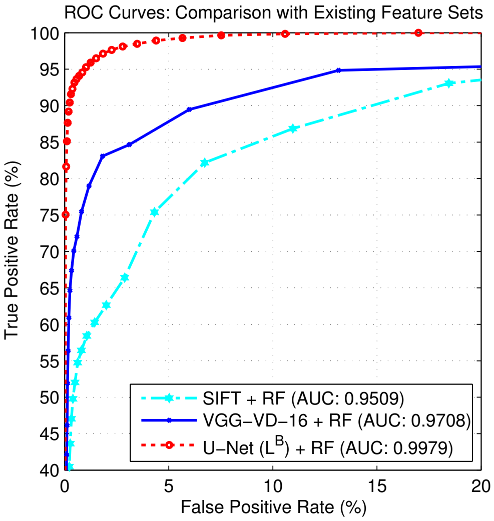


Fig. 9. ROC curves plotted using the results of random forests [2] and the SIFT [17] features, the local convolutional features extracted from the pre-trained VGG-VD-16 [22] and the features extracted from the pre-trained 3-level U-Net [21] for weld image segmentation. (Only the top-left part of each curve is shown. See main text for details).

4.4 Defect Detection

Figure 10 shows the FROC curves (proportion of defects detected vs. number of false positives per image) obtained using a hand-crafted method [25], the end-to-end trained U-Net with softmax, the U-Net features plus random forests, and the Haar-like features [24] plus random forests. We also compared the proposed MSER [18] based defect detection method with that implemented by thresholding the probability image for the two U-Net approaches. It can be seen that (1) the MSER-based defect detection performed better than the thresholding method; and (2) our approach (U-Net+RF) produced superior results to the other methods. Specifically, 84.5% of defects were detected when in average 2.4 false positives were detected per image. In contrast, the two digits are 83% and 2.7 for the Haar-like features plus random forests that Dong et al. [7] used. On the other hand, 72.4% of defects were detected using our approach with in average 0.5 false positives were obtained per image; while only 65.5% of defects were detected by the Haar-like features plus random forests. A more traditional image processing technique proposed by Wang and Liao [25] achieves much worse results; 34.5% of defects were detected when in average 31.2 false positives per image were generated.

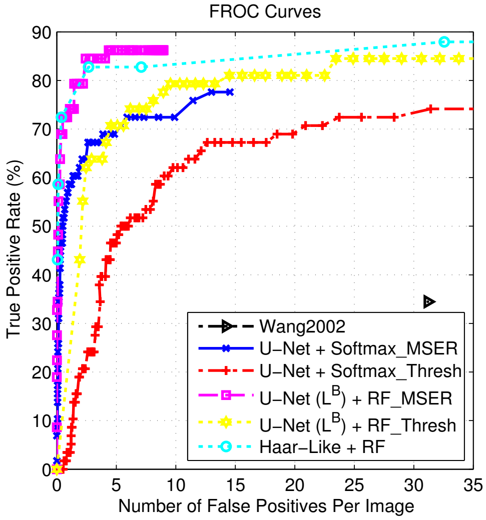


Fig. 10. The FROC curves obtained using four methods: a hand-crafted method [25], the end-to-end trained U-Net with softmax [21], the features extracted from the pre-trained U-Net plus random forests [2], and the Haar-like features [24] plus random forests. (See main text for details).

5 Conclusions

We described an approach for detecting rare, small abnormalities in imaging data, with a focus on defects visible in X-ray images of welds. The method augments U-Net with a Random Forest (RF), with the RF being trained to classify each pixel using features extracted from the U-Net. End-to-end training of U-Net leads to useful features, but the RF can make better use of them when only relatively small numbers of training examples are available. Instead of thresholding the probability map produced by the classifier, we identify defect candidates using the Maximally Stable Extremal Region (MSER) detector. This uses the higher level characteristics contained in connected components rather than simple pixel-wise thresholding. Some false positive regions can be removed by studying the pixel probabilities across the regions. The best performance obtained using our approach was superior to a variety of alternatives tested.

Although our approach was restricted by the limited number of training data, it still produced promising results. While all the learning based methods explored are likely to benefit from larger training sets, it is useful to identify techniques which will work well when few examples are available.

Acknowledgement.

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) (No. EP/L022125/1). The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

1. Boaretto, N., Centeno, T.: Automated detection of welding defects in pipelines from radiographic images dwdi. NDT & E International **86**, 7–13 (2017)
2. Breiman, L.: Random forests. Mach Learn. **45**(1), 5–32 (2001)
3. Chen, F., Jahanshahi, M.R.: Nb-cnn: Deep learning-based crack detection using convolutional neural network and nave bayes data fusion. IEEE Transactions on Industrial Electronics **65**(5), 4392–4400 (2018)
4. Chen, J., Banerjee, S., Grama, A., Scheirer, W., Chen, D.: Neuron segmentation using deep complete bipartite networks. LNCS **10434**, 21–29 (2017)
5. Ciresan, D., Gambardella, L., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images (2012), proc. NIPS, 2852–2860
6. Dong, X., Dong, J., Zhou, H., Sun, J., Tao, D.: Automatic chinese postal address block location using proximity descriptors and cooperative profit random forests. IEEE Transactions on Industrial Electronics **65**(5), 4401–4412 (2018)
7. Dong, X., Taylor, C.J., Cootes, T.F.: Automatic inspection of aerospace welds using x-ray images (2018), proc. International Conference on Pattern Recognition
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation (2014), proc. IEEE Conference on Computer Vision and Pattern Recognition

9. GoldinParis, D., Kanellakis, C.: On similarity queries for time-series data: Constraint specification and implementation (1995), proc. International Conference on Principles and Practice of Constraint Programming, 137-153
10. Hariharan, B., Arbelez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization (2015), proc. IEEE Conference on Computer Vision and Pattern Recognition
11. Kehoe, A., Parker, G.: An intelligent knowledge based approach for the automated radiographic inspection of castings. NDT & E International **25**(1), 23–36 (1992)
12. Kingma, D., Adam, J.: A method for stochastic optimization (2017), arXiv:1412.6980
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
14. Liao, T., Li, Y.: An automated radiographic ndt system for weld inspection: Part ii flaw detection. NDT & E International **31**(3), 183–192 (1998)
15. Lindner, C., Bromiley, P., Ionita, M., Cootes, T.: Robust and accurate shape model matching using random forest regression-voting. IEEE Trans. PAMI **37**(9), 862–1874 (2015)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2015), proc. IEEE Conference on Computer Vision and Pattern Recognition, 3431-3440
17. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
18. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions (2002), proc. British Machine Vision Conference, 384-396
19. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition (2014), proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 512-519
20. Ren, R., Hung, T., Tan, K.: A generic deep-learning-based approach for automated surface inspection. IEEE Transactions on Cybernetics **48**(3), 929–940 (2018)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. LNCS **9351**, 234–241 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale visual recognition (2015), proc. International Conference on Learning Representations
23. Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation (2017), arXiv:1708.06020
24. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features (2001), proc. CVPR, 511-518
25. Wang, G., Liao, T.: Automatic identification of different types of welding defects in radiographic images. NDT & E International **35**, 519–528 (2002)
26. Wang, Y., Sun, Y., Lv, P., Wang, H.: Detection of line weld defects based on multiple thresholds and support vector machine. NDT & E International **41**, 517–524 (2008)
27. Yazid, H., Arof, H., Yazid, H., Ahmad, S., Mohamed, A., Ahmad, F.: Discontinuities detection in welded joints based on inverse surface thresholding. NDT & E International **44**, 563–570 (2011)
28. Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D., Chen, D.: Deep adversarial networks for biomedical image segmentation utilizing unannotated images. LNCS **10435**, 408–416 (2017)
29. Zhao, X., He, Z., Zhang, S.: Defect detection of castings in radiography images using a robust statistical feature. J. Opt. Soc. Am. A **31**(1), 196–205 (2014)