2018 IEEE Winter Conference on Applications of Computer Vision

A compact convolutional neural network for textured surface anomaly detection

Domen Rački[1], Dejan Tomaževič[1,2] and Danijel Skočaj[3]
[1]Sensum, Computer Vision Systems
[2]University of Ljubljana, Faculty of Electrical Engineering
[3]University of Ljubljana, Faculty of Computer and Information Science
domen.racki@sensum.eu, dejan.tomazevic@fe.uni-lj.si, danijel.skocaj@fri.uni-lj.si

Abstract

Convolutional neural methods have proven to outperform other approaches in various computer vision tasks. In this paper we apply the deep learning technique to the domain of automated visual surface inspection. We design a unified CNN-based framework for segmentation and detection of surface anomalies. We investigate whether a compact CNN architecture, which exhibit fewer parameters that need to be learned, can be used, while retaining high classification accuracy. We propose and evaluate a compact CNN architecture on a dataset consisting of diverse textured surfaces with variously-shaped weakly-labeled anomalies. The proposed approach achieves state-of-the-art results in terms of anomaly segmentation as well as classification.

1. Introduction

Visual inspection systems play a vital role in defect, i.e., anomaly detection during manufacturing processes in order to ensure that the end-product is defect-free. In the domain of automated visual-inspection of surface anomalies, such as on steel surfaces [12, 14, 3, 22], textured fabrics [2, 13] or wooden surfaces [19], modeling complexity increase with increasing surface complexity. The appearance of anomalies, such as cracks, dents, smudges and impurities, varies in terms of pixel intensities, geometrical constraints and visual appearances as a whole. For some real-world computer vision problems, engineering features with enough complexity to model certain underlying patterns proves a devious task. The ability to acquire features which describe the problem in an automated manner proves indispensable in such cases. Deep learning, i.e., convolutional neural networks [10] solve this central problem in representation learning by building complex features out of simpler features.

Taking into account the problem domain of automated visual-inspection of surfaces, i.e., textured surfaces with present anomalies, we propose a compact convolutional neural network architecture for surface anomaly detection. In general, this is in stark contrast to architectures such as [24] or [21] which exhibit a substantial number of parameters that need to be learned. We argue that since the underlaying anomaly structures and patterns are limited, contrary to objects in datasets such as ImageNet, large architectures are not necessarily needed in order to successfully learn underlying anomaly patterns. In the scope of visual inspection, automated detection systems should provide a localization, i.e., segmentation of potentially abnormal, i.e., defective areas on a given surface. This provides somewhat of an insight for a human domain expert as to what the network considers as an anomaly.

The main contribution of this paper is a compact network architecture for automatic surface anomaly segmentation and detection. For a given example, the network outputs an anomaly segmentation and assigns a classification score to it. The classification score can be interpreted as the network's confidence that the example contains an anomaly. The network is capable of learning anomaly representations from a small set of coarsely labelled training examples. We evaluate the proposed network on different training setups on a dataset consisting of diverse textured surfaces with intra-class background and anomaly variations.

The remainder of the paper is structured as follows. In Section 2 we provide an overview of related work, followed by the description of the proposed architecture for anomaly detection in Section 3. The experimental setup is described in Section 4 while results are presented in Section 5. We conclude with the discussion in Section 6.

2. Related work

Classical approaches in anomaly detection on object surfaces follow more or less the same paradigm, i.e., a classifier such as SVM, LDA, PCA, Hough transform, decision trees or KNN trained on feature descriptors obtained from preprocessed images. The preprocessing stage is crucial in such approaches as it ensures that the problem is well-conditioned for the process of hand-engineering suit-

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00150

able features. Commonly used non-deep learning based anomaly detection approaches [26, 9] include: (i) Filtering approaches based on Wavelet transform [4], independent component analysis [19] and Gabor filter [28]; (ii) Structural approaches based on morphological operators [13] and edge detection [18]; (iii) Model based approaches such as the Hidden Markov model [17] and autoregressive models [1]; (iv) Statistical approaches based on histograms [1], co-occurrence matrices [5] and autocorrelation [7]. Deep learning highly contrast these approaches by performing automated feature learning instead of hand-crafting suitable and at times sub-optimal features. Deep learning, i.e., convolutional neural networks have proven superior in tasks where hand designing features proves a difficult task.

Early work on utilizing a CNN for surface anomaly detection can be found in [14]. The motivation arises from the aforementioned difficulty, where even domain specialists struggle to devise accurate rules based on geometrical and shape features for certain defects. With deep learning, the classification error is significantly reduced compared to classical approaches with a classifier trained on feature descriptors. These included a Multi Layer Perceptron (MLP) and SVM with RBF classifiers trained on features obtained via HOG, PHOG, rotation invariant measure of local variance, and Local Binary Patterns (LBP, LBP-Fourier). Taking new deep learning research insights into account, an overview of different CNN design heuristics for industrial inspection is presented in [24]. The paper examines the impact of different hyper-parameter settings with respect to the accuracy for anomaly detection. Evaluation is performed on an artificial dataset, as shown in Figure 3, comprised of diverse surfaces on which the goal is to detect anomalies. The dataset consists of artificially generated images which imitate diverse textured surfaces with variously shaped anomalies. Recent approaches such as [15, 24, 20, 23] are evaluated on the aforementioned dataset, making it suitable for benchmarking new approaches. Other work on utilizing deep learning for anomaly detection includes feature extraction for the detection of rail surface anomalies from automated video recordings [3] and learning rail surface anomalies from photometric stereo images [22]. The images, made visible in a photometric dark-field setup, depict differently colored light-sources which illuminate rail surfaces from different and constant directions. Anomaly detection on X-ray images of automotive metallic components is researched in [16].

The aforementioned work showcases the feasibility of utilizing deep learning for the problem of anomaly detection on different surfaces. In this paper we expand on previous deep learning research and focus on textured surface anomaly detection. Textured surfaces are common in industrial inspection where a crucial aspect is the ability to adequately distinguish anomalies from background patterns.

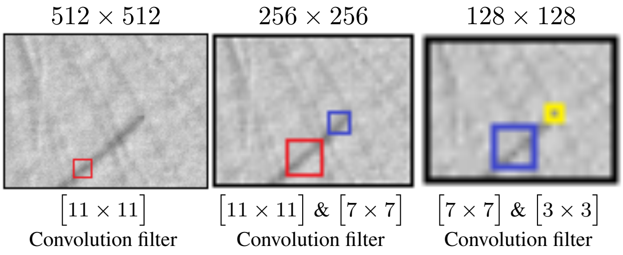

Figure 1. Convolutional filter dimensions with respect to the zoomed anomaly example from Surface 4 show in Figure 3. The first image depicts the original input size of 512×512, while each subsequent image displays the size after subsampling. The red square represents a filter of size 11×11, the blue square a filter of size 7×7 and the yellow square a filter of size 3×3 pixels.

3. Architecture design

Deep learning based approaches have proven superior to non-deep learning based methods in a variety of different tasks, ranging from detection [14] to segmentation [11]. However, in the global scope of things, a more-or-less common practice is to adapt an existing architecture, such as VGG [21], AlexNet [8], for a specific task in a way that utilizes the pre-trained weights of the lower layers on large datasets and merely adapts the weights of a few upper layers for a new problem domain. In general this approach is cumbersome when dealing with specific problem domains such as anomaly detection on surfaces. This is due to the fact that large architectures implicitly exhibit the need for large training sets which are usually not available in industrial environments, as the acquisition of defective examples presents a costly and impractical task. As such, certain criteria for the network architecture can be identified. A network designed for an inspection system should: (i) Be compact, i.e., be able to learn potential anomaly detection from a handful of defective examples. (ii) Be robust, i.e., within a similar problem domain a network should require merely slight, if any, hyperparameter adjustments. (iii) Be explainable, i.e., be able to provide visual localization and classification explanation to a human domain expert.

Considering the aforementioned criteria, we propose a compact convolutional network architecture comprised of a segmentation and classification stage. The task of the classification stage is to assign a given example a score which can be interpreted as the network's confidence that a given example contains an anomaly. The task of the segmentation stage is to provide visual anomaly localization and classification explanation to a human domain expert. The proposed network architecture is shown in Figure 2.

The segmentation part of the network consists of three convolution blocks, and each block consists of three convolution layers. In general the number of features increases by a factor of two in each convolution block. Conversely the filter size in each block is decreased. When choosing a filter size, we can make some valid assumptions about the problem at hand and translate these into deep learning architecture design choices. The filters should enclose part of the anomaly that should be detected, as shown in Figure 1. Here we choose filters in such a way that the ratio between two subsequent filter sizes is kept during subsampling stages. This ensures that the filter continues to enclose part of the anomaly throughout the network. The actual representation within the network will differ from the depictions, however, this is merely to illustrate the reasoning. In general the coarse factor-of-two feature count step size combined with a stride-of-two convolution at the beginning of a block and a step-wise filter size reduction keeps the parameter count from increasing manifold, achieving compactness.

The classification part of the network relies on the segmentation part. Classification scoring of a given example is achieved via a maximum and average global pooling combination from the segmentation layer (*SegLayer*) and compression layer (*CompLayer*) as illustrated in Figure 2. The segmentation layers is inspired by fully convolutional networks [11] and merely provides the segmentation output from one layer above. The compression layer on the other hand serves to compress the activation volume from one layer above. This reduces the number of parameters from which the classification score is estimated and robustifies the classification score. The usage of maximum and average global pooling proves robust in cases where for example a larger defect-free area would be segmented as a defective region. The maximum pooling alone would fail to see any difference between this and an example where merely a small defective area would be segmented. This is the underlying reason why we additionally perform average pooling, as this would exhibit a difference in the aforementioned example.

4. Experimental setup

All experiments, that is, for all training-setups and all surfaces, are ran with a fixed set of network hyperparameters. Network hyperparameters used and the network architecture are shown in Figure 2. Given an input image of size 512 × 512 pixels, our network outputs a segmentation map of size 128 × 128 pixels. For all layers within the network the ReLU activation function is used after which batch normalization is applied. The only exceptions are the *SegLayer* and *S-neuron* where we use the linear and sigmoid activation function respectively. All network weights are initialized with a normal distribution centered around zero, as proposed in [6]. Network training is performed in two stages. In the first stage, i.e., *Segmentation stage* we train the anomaly segmentation step of the network for 25 epochs. In the second stage, i.e., *Classification stage* we train the classification step of the network for 10 epochs. The two stage training is essential. In the first stage all classification layers are frozen and merely the segmentation layers are learned. Conversely, in the second stage all the segmentation layers are frozen and merely the classification layers are trained. This ensures that the classification will be trained on meaningful segmentation representations. The two training stages are outlined in Figure 2. In both cases the network is learning a regression value from either $[-1, 1]$, which is assigned to each pixel in the segmentation step, or $[0, 1]$ which is assigned to a single example in the classification step. During the segmentation stage training we minimize the mean squared-error loss function, i.e.,

$$\mathcal{L}_S = \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} ||x_i^{\langle j \rangle} - \hat{x}_i^{\langle j \rangle}||^2 \tag{1}$$

where n denotes the number of examples, p the number of pixels, x_i the annotated pixel value and $\hat{x}_i$ the predicted pixel value. In the classification stage the binary cross-entropy loss function, i.e,

$$\mathcal{L}_C = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \tag{2}$$

is minimized, where n denotes the number of examples, y_i the ground truth example annotation and $\hat{y}_i$ the example regression prediction. Both functions are minimized using the Adadelta optimizer [27] with the parameters left at default values as suggested in the paper.

4.1. The DAGM dataset

We evaluated our proposed approach on the dataset for Industrial Optical Inspection [1], which consists of artificially generated textured surfaces. As can be seen in Figure 3 the dataset is comprised of ten diverse surface classes with diverse anomalies emulating cracks, dents, smudges and impurities, each generated by a different texture and defect model. We refer to a given example as *positive* if it contains an anomaly and *negative* if the example is anomaly-free.

Table 1 depicts the distribution of training and testing examples over the dataset. The entire dataset consists of 8050 train examples of which 1046 contain anomalies, and 8050 test examples of which 1054 contain anomalies. If a given surface of size 512x512 pixels contains an anomaly, it contains exactly one weakly labeled anomaly on the background texture. Weak labels are provided in form of ellipses which roughly indicate the defective area on a given example, but also include defect-free areas to some extent, as shown in Figure 3. Although the entire anomaly is contained within the encircling ellipse, a significant portion of the regular surface is encircled as well. Consequently, a

[1] https://hci.iwr.uni-heidelberg.de/node/3616

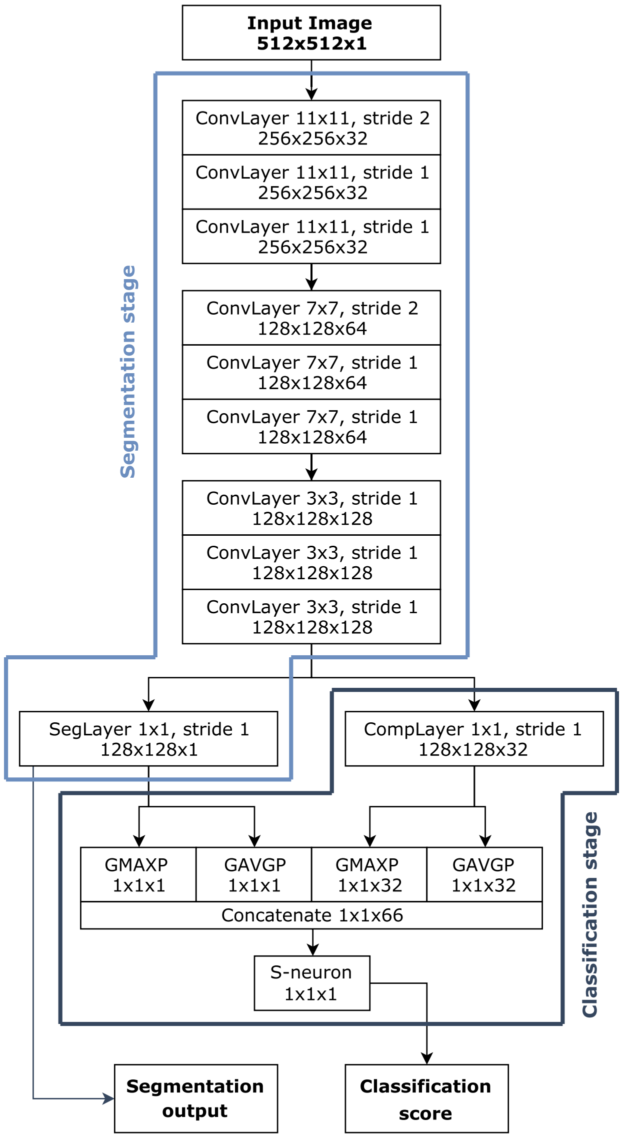

Figure 2. The proposed CNN-architecture whit approximately 1.1M parameters which outputs a segmentation and a classiciation score for a given example. The abbreviations ConvLayer, SegLayer and CompLayer stand for convolutional, segmentation and compression layer respectively. Abbreviations GMAXP and GAVGP stand for global maximum pooling and global average pooling, and S-neuron stands for scoring neuron.

number of image pixels are falsely labelled, which can affect the learning process. However, this problem is to be expected in many real world situations since precise annotations of surface anomalies are difficult and costly to obtain. In general, datasets intended for industrial inspection systems are hard to come by, for which the underlying reasons are non-disclosure agreements which serve to prevent the disclosure of company secrets such as image acquisition or other crucial processes which ensure a competitive advantage.

Surface	Train examples		Test examples	
	Positive	Negative	Positive	Negative
1	79	496	71	504
2	66	509	84	491
3	66	509	85	490
4	82	493	68	507
5	70	505	81	494
6	83	492	67	508
7	150	1000	150	1000
8	150	1000	150	1000
9	150	1000	150	1000
10	150	1000	150	1000

Table 1. Distribution of train and test examples over the dataset.

Name	Training examples used
Pos	only positive examples
PosNeg	positive and negative examples
Pos-aug	*Pos* with data augmentation
PosNeg-aug	*PosNeg* with data augmentation
Positive examples	Examples with defects
Negative examples	Defect-free examples

Table 2. Different training setups.

4.2. Training setup

The acquisition of defective examples presents a costly, and at times impossible, task in real world production scenarios, while defect-free examples are usually available in large quantities. A network designed for an inspection system should thus be able to learn potential anomaly detection from a handful of defective examples. We evaluate our proposed network on different training setups, as listed in Table 2. These consists of training the network on merely the available positive examples, denoted *Pos*, and positive and negative examples denoted *PosNeg*. However, as a small number of examples can prove insufficient since these may not provide a canonical overview of the problem domain, we also perform training on augmented positive examples, i.e., *Pos-aug* and *PosNeg-aug*. Here each given positive training example is rotated for 180° and mirrored around the horizontal and vertical axis. For each positive training example we thus obtain the original plus additional three augmented examples, increasing the number of positive examples in the training set four-fold and leaving the number of negative examples unchanged.

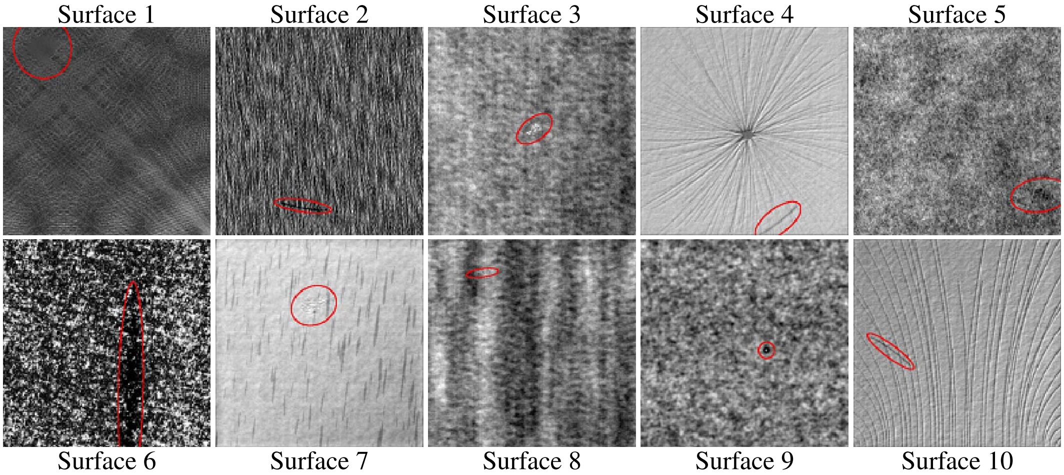

Figure 3. A snapshot of the diverse textured surfaces in the dataset. Each surface class exhibits additional intra-class variation of the background texture. Red ellipses present coarse surface anomaly labeling, i.e., weakly labeled ground truth annotations as these include areas which do not correspond to anomalies.

5. Results

We evaluate the performance of our network in terms of the true positive rate, i.e., positives that are correctly identified as positives and true negative rate, i.e., negatives that are correctly identified as negatives. We also evaluate the performance of our network in terms of the absolute number of misclassified test examples. Table 3 depicts the performance of our network on different training setups. As can be seen, our network when trained on the training setup *PosNeg-aug* fails to detect merely one positive test example, while maintaining a high negative, i.e., defect-free example detection accuracy. Although abnormal regions are represented merely by a small ratio of pixels, the overall performance is noticeably increased with augmented negative and the addition of positive examples.

A crucial aspect of any detection system is the detection robustness, i.e., the effect on performance when varying the classification threshold. Figure 4 depicts the detection robustness when varying the classification threshold on the test set. As can be seen, the network exhibits high robustness to classification threshold variation when trained on the training setup *PosNeg-aug*. Table 4 shows the performance compared to a deep learning based method proposed in [24] and other non-deep learning based methods [20, 25, 23]. Our network outperforms all methods in terms of the the detection of defective examples, while maintaining a high defect-free example detection.

The high-level performance aspect of the network is promising, however a qualitative analysis of the segmentation outputs provides further insights. We perform a qualitative analysis since a detailed quantitative analysis cannot be performed due to the available weak anomaly annotations, i.e., non-pixel-level annotations. Figure 5 depicts the performance of the network in terms of segmentation, i.e., emphasizing annotated regions and suppressing the background; and classification, i.e., assigning a score to a given example. The performance is increases with the training setup *PosNeg-aug* with which background-artifacts are strongly suppressed. As mentioned, our network fails to detect one positive example. This false negative detection is depicted in Figure 7 and is due to a poor segmentation of the anomaly from the background. In general, false negative examples are scored with a low value in the case when the segmentation performance is insufficient, not necessarily in terms of anomaly segmentation but also in terms of background suppression. This is the case for the training setup *Pos*, where a positive example with a sufficient segmentation result was classified as false negative, due to weak background suppression on other negative examples. Figure 6 depicts a false positive detection example. Although the training setup *PosNeg-aug* exhibits better overall performance, in the given case the network is unable to suppress the textured background and consequently outputs a high classification score.

Concluding from the segmentation examples, data augmentation and the addition of negative training examples play a vital role in suppressing noisy background segmentations. In general, the segmentation step is crucial not only for the classification step, but also as it provides a human domain expert with the visual explanation of why a given example was classified as either defective or defect-free. Furthermore, we find that our network is robust to weak annotations. Even though the provided annotations are weak in the sense that they roughly indicate a defective area but also include defect-free areas to some extent, the network manages to perform efficient anomaly segmentation.

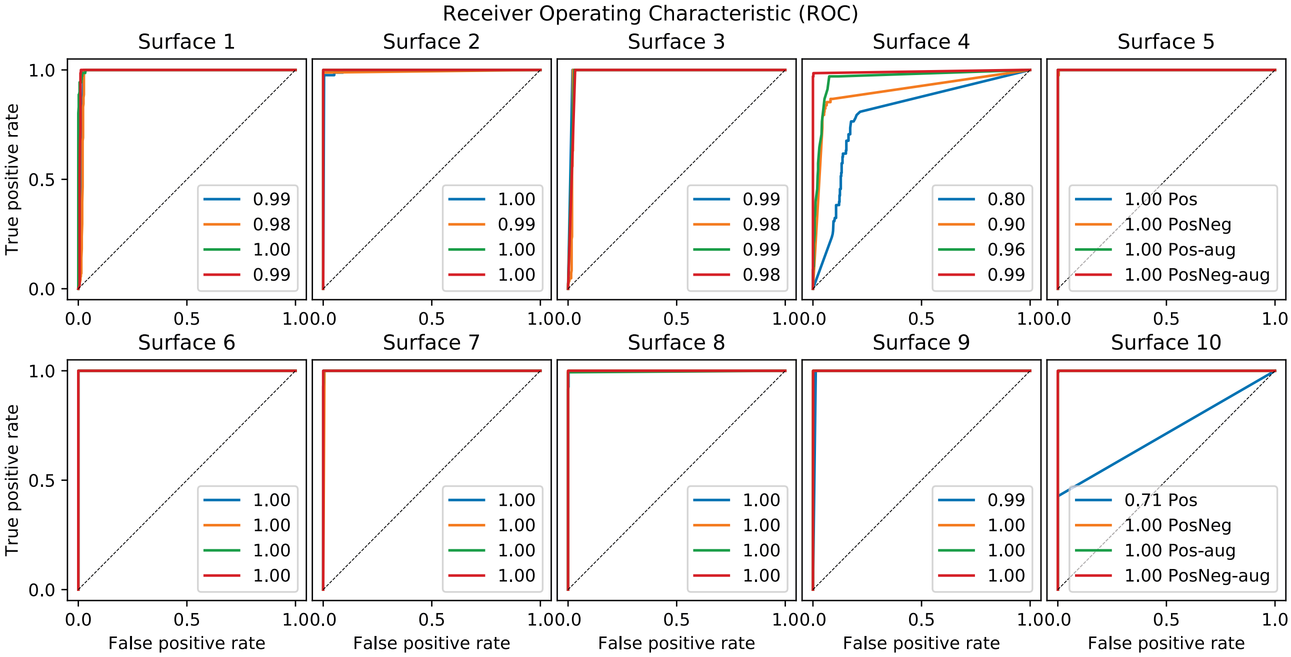

Figure 4. Receiver Operating Characteristic (ROC) curve for each surface class as shown in Figure 3. The curves are obtained by thresholding anomaly score predictions from the testing set for each given example from 0 to 1, and classifying an example as positive if the score prediction exceeds the threshold. The numbers below the curves indicate the area under the curve (AUC). The figures indicate the likelihood that an example with an anomaly will be classified correctly (True positive rate) vs. the likelihood that an example without an anomaly will be classified falsely (False positive rate) when learning on different training setups.

Surface	*Pos*		*PosNeg*		*Pos-aug*		*PosNeg-aug*	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
1	100 [0]	97.6 [12]	100 [0]	96.4 [18]	98.6 [1]	97.8 [11]	100 [0]	98.8 [6]
2	97.6 [2]	99.6 [2]	98.8 [1]	99.6 [2]	100 [0]	99.8 [1]	100 [0]	99.8 [1]
3	100 [0]	97.8 [11]	100 [0]	97.1 [14]	100 [0]	96.7 [16]	100 [0]	96.3 [18]
4	0.0 [68]	100 [0]	77.9 [15]	95.7 [22]	97.1 [2]	88.4 [59]	98.5 [1]	99.8 [1]
5	100 [0]	99.8 [1]	100 [0]	99.6 [2]	100 [0]	100 [0]	100 [0]	100 [0]
6	100 [0]	99.8 [1]	100 [0]	100 [0]	100 [0]	100 [0]	100 [0]	100 [0]
7	100 [0]	99.7 [3]	100 [0]	98.9 [11]	100 [0]	98.4 [16]	100 [0]	100 [0]
8	100 [0]	99.1 [9]	100 [0]	99.9 [1]	99.3 [7]	99.9 [1]	100 [0]	100 [0]
9	100 [0]	98 [20]	100 [0]	100 [0]	100 [0]	99.9 [1]	100 [0]	99.9 [1]
10	42.7 [86]	100 [0]	100 [0]	99.7 [3]	100 [0]	99.6 [4]	100 [0]	100 [0]
Σ	[156]	[59]	[16]	[73]	[4]	[109]	[1]	[27]

Table 3. Classification performance of our proposed CNN architecture on different training setups as listed in Table 2. The numbers in the square brackets denote the absolute number of examples that were misclassified for each surface, while the bottom numbers indicate the total number of misclassified examples w.r.t. the total number of examples in the testing set as shown in Table 1, which amount to 1056 for TPR and 6994 for TNR.

6. Conclusion

We propose a compact convolutional neural network for the segmentation and detection of surface anomalies. We performe a number of experiments on a dataset consisting of diverse textured surfaces with variously-shaped weakly-labeled anomalies in order to evaluate the proposed approach. We achieve state-of-the-art results in terms of anomaly segmentation as well as classification. The proposed network is compact in terms of parameter count, proves robust across diverse surface textures and provides visual localization and classification explanation for a human domain expert via the segmentation and classification score output.

Surface	*Ours*		*Weimer et. al. [24]*		*Statistical feat. [20]*		*SIFT and ANN [25]*		*Weibull feat. [23]*	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
1	100	98.8	100	100	99.7	99.4	100	98.9	98.0	87.0
2	100	99.8	97.3	100	80.0	94.3	91.3	95.7	-	-
3	100	96.3	100	95.5	100	99.5	100	98.5	100	99.8
4	98.5	99.8	98.7	100	96.1	92.5	-	-	-	-
5	100	100	100	98.8	96.1	96.9	100	98.2	100	97.2
6	100	100	99.5	100	96.1	100	100	99.8	100	94.9
7	100	100	-	-	-	-	-	-	-	-
8	100	100	-	-	-	-	-	-	-	-
9	100	99.9	-	-	-	-	-	-	-	-
10	100	100	-	-	-	-	-	-	-	-

Table 4. Classification performance of our proposed CNN architecture vs others.

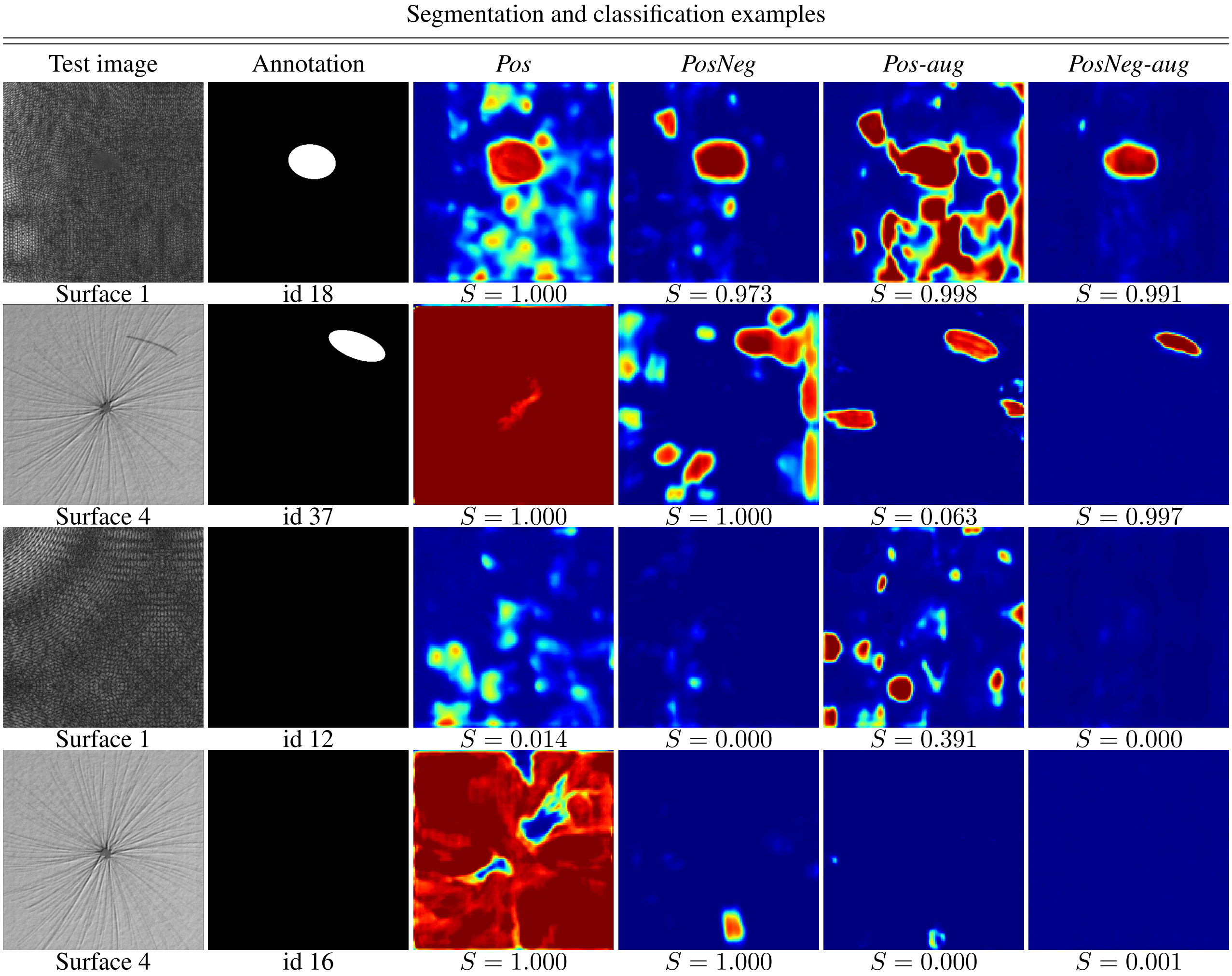

Figure 5. Results w.r.t. emphasizing annotated regions, suppressing the background and assigning a classification score to a given example.

7. Acknowledgement

This work was supported by the Ministry of Economic Development and Technology (MGRT), Republic of Slovenia; the European Union, European Regional Development Fund (ERDF) under grant 631-63/2017/1; and Sensum, Computer Vision Systems.

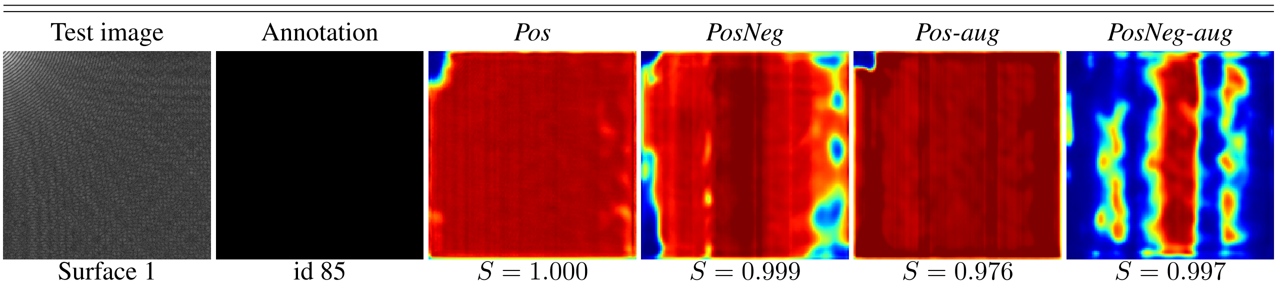

Figure 6. False positive segmentation and classification example.

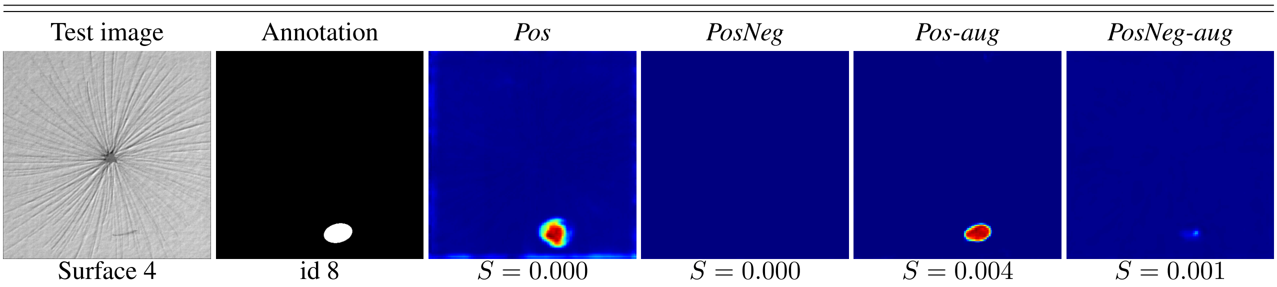

Figure 7. The only false negative segmentation and classification example.

References

[1] H.-G. Bu, X.-B. Huang, J. Wang, and X. Chen. Detection of fabric defects by auto-regressive spectral analysis and support vector data description. *Textile Research Journal*, 80(7):579–589, 2010.

[2] S. Chen, J. Feng, and L. Zou. Study of fabric defects detection through gabor filter based on scale transformation. In *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pages 97–99. IEEE, 2010.

[3] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter. Deep convolutional neural networks for detection of rail surface defects. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2584–2589. IEEE, 2016.

[4] M. Ghazvini, S. Monadjemi, N. Movahhedinia, and K. Jamshidi. Defect detection of tiles using 2d-wavelet transform and statistical features. *World Academy of Science, Engineering and Technology*, 49:901–904, 2009.

[5] R. M. Haralick, K. Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, 3(6):610–621, 1973.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] E. Hoseini, F. Farhadi, and F. Tajeripour. Fabric defect detection using auto-correlation function. *International Journal of Computer Theory and Engineering*, 5(1):114, 2013.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[9] A. Kumar. Computer-vision-based fabric defect detection: A survey. *IEEE Transactions on Industrial Electronics*, 55(1):348–363, Jan 2008.

[10] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[12] A. M. Luiz, L. P. Flávio, and E. A. Paulo. Automatic detection of surface defects on rolled steel using computer vision and artificial neural networks. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pages 1081–1086. IEEE, 2010.

[13] K.-L. Mak, P. Peng, and K. Yiu. Fabric defect detection using morphological filters. *Image and Vision Computing*, 27(10):1585–1592, 2009.

[14] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout. Steel defect classification with max-pooling convolutional neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.

[15] C. Mera, M. Orozco-Alzate, J. Branch, and D. Mery. Automatic visual inspection: An approach with multi-instance learning. *Computers in Industry*, 83:46 – 54, 2016.

[16] D. Mery and C. Arteta. Automatic defect recognition in x-ray testing using computer vision. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1026–1035, March 2017.

[17] A. Mukherjee, S. Chaudhuri, P. K. Dutta, S. Sen, and A. Patra. An object-based coding scheme for frontal surface of defective fluted ingot. *ISA transactions*, 45(1):1–8, 2006.

[18] F. S. Najafabadi and H. Pourghassem. Corner defect detection based on dot product in ceramic tile images. In *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*, pages 293–297. IEEE, 2011.

[19] W. Polzleitner. Defect detection on wooden surface using gabor filters with evolutionary algorithm design. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 1, pages 750–755. IEEE, 2001.

[20] B. Scholz-Reiter, D. Weimer, and H. Thamer. Automated surface inspection of cold-formed micro-parts. *CIRP Annals*, 61(1):531 – 534, 2012.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] D. Soukup and R. Huber-Mörk. Convolutional neural networks for steel surface defect detection from photometric stereo images. In *International Symposium on Visual Computing*, pages 668–677. Springer, 2014.

[23] F. Timm and E. Barth. Non-parametric texture defect detection using weibull features. *IS&T SPIE Electronic Imaging. International Society for Optics and Photonics*, 2011.

[24] D. Weimer, B. Scholz-Reiter, and M. Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals-Manufacturing Technology*, 2016.

[25] D. Weimer, H. Thamer, and B. Scholz-Reiter. Learning defect classifiers for textured surfaces using neural networks and statistical feature representations. *Procedia CIRP*, 7(Supplement C):347 – 352, 2013. Forty Sixth CIRP Conference on Manufacturing Systems 2013.

[26] X. Xie. A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 7(3), 2008.

[27] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[28] H. Zheng, L. X. Kong, and S. Nahavandi. Automatic inspection of metallic surface defects using genetic algorithms. *Journal of materials processing technology*, 125:427–433, 2002.