Towards surface anomaly detection with deep learning

Domen Rački[1], Dejan Tomaževič[2,1], Danijel Skočaj[3]

[1]*Sensum. Computer Vision Systems. Automatic inspection of tablets and capsules, Tehnološki park 21, Ljubljana, Slovenia*
[2]*University of Ljubljana, Faculty of Electrical Engineering, Tržaška c. 25, 1000 Ljubljana, Slovenia*
[3]*University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, Ljubljana, Slovenia*
E-mail: domen.racki@sensum.eu, dejan.tomazevic@fe.uni-lj.si, danijel.skocaj@fri.uni-lj.si

Abstract

In this paper we investigate whether shallower CNN architectures, which exhibit less parameters that need to be learned, can be used in the domain of automated visual-inspection of surface anomalies, while retaining a high classification accuracy. We examine whether learning can be done merely on positive examples as this further reduces the overall computational complexity. We evaluate a shallow CNN architecture on a dataset consisting of different textured surfaces with variously-shaped weakly-labeled anomalies. We find that high classification accuracy can be achieved with shallow CNN models.

1 Introduction

Visual inspection systems play a vital role in defect, i.e., anomaly detection during a manufacturing process 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 [6], textured fabrics [1] or wooden surfaces [7], modeling complexity increase with increasing surface complexity. As such, the appearance of anomalies 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 (CNN) [4] solve this central problem in representation learning by building complex features out of simpler features. However, determining suitable network hyperparameters in general requires a somewhat stochastic search over the heyperparameter space, with the way the training data is setup additionally impacting the performance. In this paper we ran several experiments in order to determine the most promising way in which to setup the training set for the task of surface anomaly detection.

The remainder of the paper is structured as follows. In Section 2 we provide a brief overview of related work, followed by the description of of our approach in Section 3. The experimental setup is described and results are presented in Section 4. We conclude with the discussion in Section 5.

2 Related work

Early work on utilizing a CNN for surface anomaly detection can be found in [6]. 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. Authors manage to reduce the classification error by half over the classical approach with a classifier trained on feature descriptors, which included a Multi Layer Perceptron (MLP) and SVM with RBF classifiers trained on features obtained via HOG, PHOG. Taking new deep learning research insights into account, authors in [10] present an overview of different design heuristics of CNN for industrial inspection. 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 1, comprised of different surfaces on which the goal is to detect anomalies. Although the dataset consists of artificially generated images, these imitate different textured surfaces with variously shaped anomalies. Other work on utilizing deep learning for anomaly detection can be found, such as learning from photometric stereo images of rail surface anomalies, where images depict differently colored light-sources illuminating the rail surfaces from different and constant directions, made visible in a photometric dark-field setup [9]. Or the usage of deep learning for non-trivial extraction of suitable features for the detection of rail surface anomalies from raw automated video recordings [2]. The aforementioned papers showcase the feasibility of utilizing deep learning for the problem of detecting anomalies on different surfaces.

3 Our approach

Taking into account the problem domain of automated visual-inspection of surfaces, i.e., mostly textured surfaces with present anomalies, we evaluate a rather shallow CNN architecture comprised of seven convolutional layers, with approximately 7.5M parameters as shown in Table 1. In general, this is in stark contrast to other architectures, such as [10] or [8], where the depths range mostly from 11 to 19 layers, and which exhibit substantially more parameters which need to be learned, as in the case of the ”depth-3”, ”width-2” model configuration

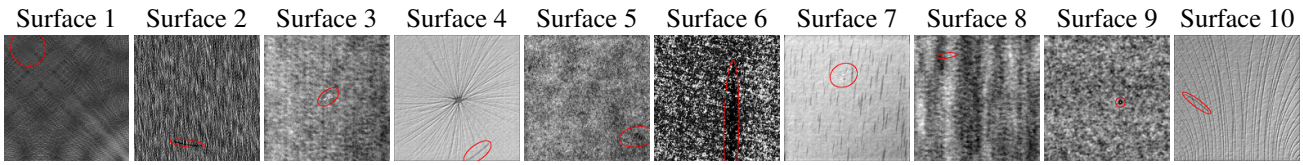

Figure 1: A snapshot of the ten different-textured surfaces in the dataset. Each surface class exhibits additional intra-class variation of the background texture. Red ellipses present coarse surface anomaly labeling, i.e., weakly labeled ground truth annotations as these include areas which do not correspond to anomalies.

proposed in [10] with 21.5M parameters. We argue that since the underlaying structures and patterns are limited, contrary to datasets such as ImageNet, very deep CNN are not necessarily needed in order to successfully learn the underlying anomaly patterns.

Additionally, given the nature of the dataset — described in Subsection 4.2 and used in [10] — we evaluate whether it is possible to learn anomaly representations merely from examples with an anomaly (positive examples), rather than from positive and negative examples (surfaces wihout anomalies). We argue that since for a given example an anomaly occupies a small portion of an image depicting a textured surface, each example provides positive and negative samples from which we can learn. Furthermore, we are dealing with unbalanced sets, as the majority of pixels correspond to negative (non anomaly) samples and merely a fraction of pixels to positive (anomaly) samples, as can be seen in Figure 1.

Our approach further differs from [10] as: I.) We adapt the final layers of our CNN to output a segmentation, i.e., a probability map of anomaly locations — as proposed in [5] — since this proves useful for labeling regions on surfaces which contain an anomaly; II.) We train CNN in a weakly supervised manner on whole images, which contrasts the patch-wise approach in [10], trained on around 1.3M generated examples. As we perform training on a significantly smaller number of examples, this additionally lowers the overall computation cost.

Table 1: The shallow 7-layer CNN model architecture. The number in each row of the table corresponds to the number of features used in that layer of our proposed architecture.

shallow-7
32
32 (stride 2)
64
64
512
512 (stride 2)
1024
Fully convolutional layer
Approximately 7.5M parameters

4 Experimental results

4.1 Experimental setup

We ran several experiments in order to determine: I.) The best way to scale ground truth labels; II.) The most promising way to setup the training set; III.) The detection accuracy when classifying the obtained probability maps by means of simple thresholding. Experiments are done with a fixed set of network parameters, i.e., convolutional kernels are fixed to a size of 3×3 pixels with a stride of one, as proposed in [8], except for the second and sixth layer with a stride of two. Given an input image of size 512×512 pixels, our network outputs a probability map of size 128×128 pixels. For all layers within the network the ReLU activation function, i.e., $f(x) = max(0, x)$ is used, except for the last layer where we use a linear activation function. All weights are initialized with a normal distribution centered around zero, as proposed in [3]. We optimize the mean-squared-error loss function for 10 epochs with the Adadelta optimizer [11], with the learning rate set to $l = 0.75$ and other parameters left at default values as suggested in the paper. CNN training was performed for each surface class separately on as little as approximately 100 negative examples, i.e., examples containing an anomaly.

4.2 The DAGM dataset

The dataset for Industrial Optical Inspection[1] consists of artificially generated textured surfaces of size 512x512 pixels. There are ten different surface classes, each generated by a different texture and defect model. The entire dataset consists of 8050 train examples of which 1046 contain anomalies, and 8050 test examples of which 1054 contain anomalies. For each surface class, there are approximately 100 images with present anomalies. If a given surface 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 a defective are on a given example, as shown for each surface class in Figure 1.

4.3 Results

4.3.1 Ground truth scaling

We examine the effect of scaling ground truth labels, as shown in Figure 3, either to the interval of $[-1, 1]$ or $[0, 1]$, where -1 or 0 denote the surface background pixels, and 1 the pixels which correspond to an anomaly. Figure 2 depicts the training set ROC curves and AUC values. These are obtained by thresholding the probability maps, obtaind with the CNN trained on each surface separatly, from 0 to 1 — corresponding to the highest expected probability — with a step size of 0.01. The ROC curves in Figure 2 for Surfaces 4, 8 and 9 clearly show that in our given case the best practice is to scale the ground truth to the interval of $[-1, 1]$.

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

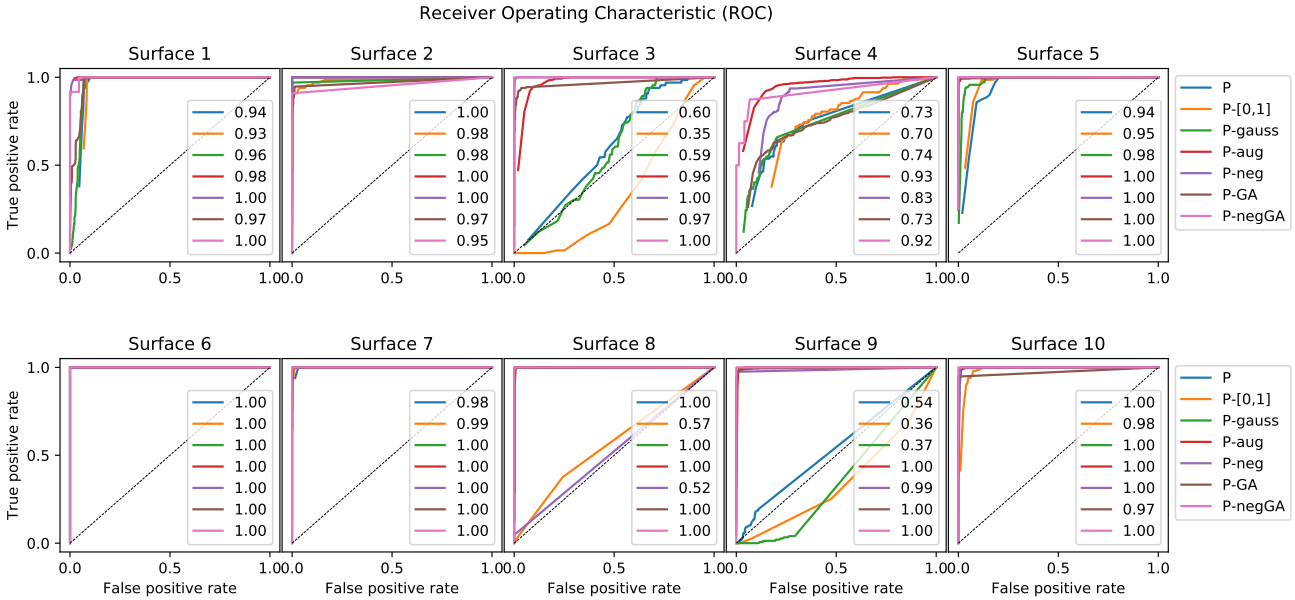

Figure 2: Receiver Operating Characteristic (ROC) curve for each textured surface as shown in Figure 1. The curves are obtained by means of simple thresholding each probability map for each surface from 0 to 1, and classifying an example as abnormal if at least one pixel is above the threshold. The numbers below the curves indicate the area under the curve (AUC) for each test. The figures indicate the probability that an example with an anomaly will be classified correctly (True positive rate) vs. the probability that an example without an anomaly will be classified falsely (False positive rate). The legend is described in Table 2.

4.3.2 *Training on different setups*

Surface anomalies are only weakly labelled in the DAGM dataset, as can be observed in 1. It is guaranteed that the entire anomaly is contained within the encircling ellipse, however also a significant portion of the regular surface is labelled as well. Consequently, many image pixels are wrongly labelled as problematic, which affects the learning process. This problem is to be expected in many real world situations since very precise annotations of surface anomalies are very difficult (and costly) to obtain. To reduce this problem we weight the values inside the ellipse with Gaussian kernel, emphasizing the values in the center of the ellipse (where the actual anomaly is most probably located), while decreasing the values of the labelled pixels at the border of the ellipse, since these pixels are most probably not affected by the anomaly.

We examine the effect of training our CNN on different setups, described in Table 2. These include weighing ground truth annotations with a Gaussian kernel, such that a Gaussian of width 100×100 pixels and a sigma of $\sigma = 1$ is transformed via the eigenvectors to the ellipses denoting abnormal regions, and, data augmentation, which is performed by simply rotating an example image three times for $90°$ and mirroring the image via the horizontal and vertical axis.

As can be seen from the AUC values and from the qualitative examination of the probability maps given in Figure 3 in column "CNN model activations", the clearest probability maps are obtained when either training the CNN on merely positive examples which are weighted with a Gaussian distribution and subjected to data augmentation (*P-GA*), or, training the CNN on positive and negative examples which are weighted with a Gaussian distribution and subjected to data augmentation (*P-negGA*), with the latter exhibiting the higher overall AUC value.

4.3.3 *Probability map classification*

For each surface we determine the optimal cutt-off threshold value τ_i on the training set, from the probability maps obtaind with the CNN trained on *P-negGA*. Each given example in the test set is thus thresholded at τ_i. A given example is classified as positive (having an anomaly) if there is at least one pixel which is above the determine threshold value. Results obtained with this simple thresholding procedure as presented in Table 3.

Comparing our shallow CNN performance with [10] in Table 3, we can see that our initial assumptions about the CNN depth and learning from a smaller set of examples hold true in most cases. As can be seen, we reach nearly state of the art performance with a shallow CNN architecture, trained on a significantly smaller set.

Table 2: Figure legend explanation.

Name	Description
P	Model trained on positive examples
P-[0,1]	*P* with ground truth scaled to the interval [0,1]
P-gauss	*P* with Gaussian weighted ground truth
P-aug	*P* with data augmentation
P-neg	Model trained on positive and negative examples
P-GA	*P* with Gaussian weighted ground truth and data augmentation
P-negGA	*P-neg* with Gaussian weighted ground truth and data augmentation

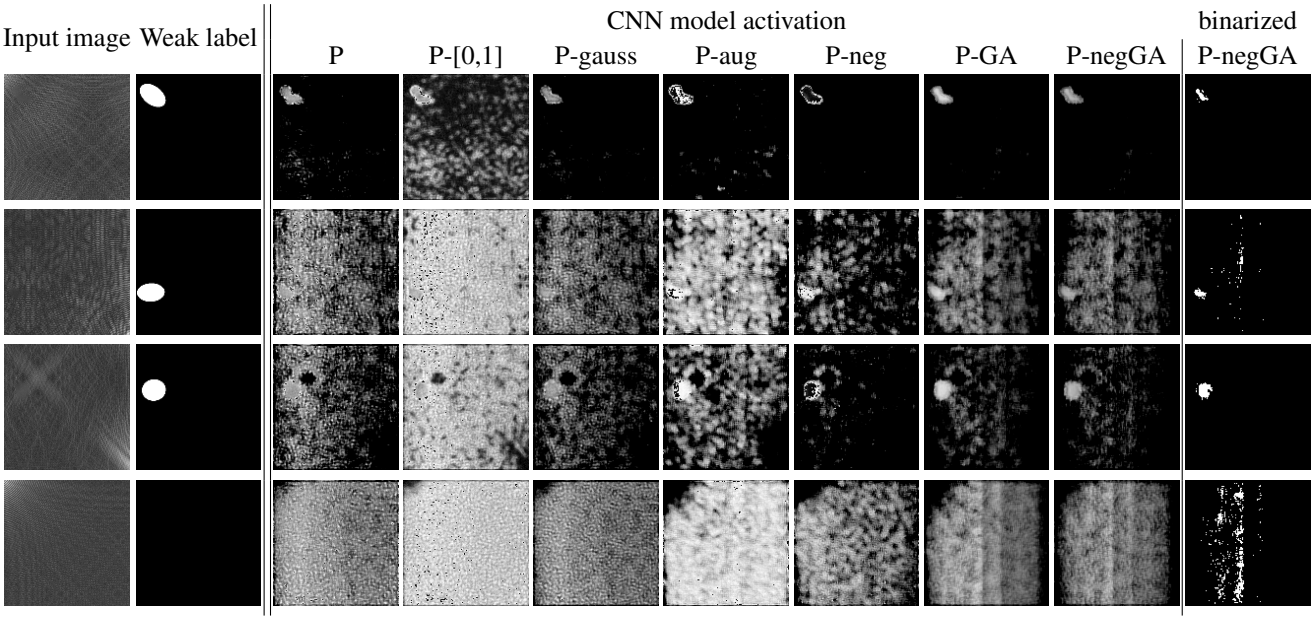

Figure 3: Activation results on different examples from textured surface no. 1. The activations depict different performances for intra-class background pattern variations w.r.t. emphasizing labeled regions and suppressing the background. The last column represents the binarized activations from the training setup *P-negGA*, thresholded at the value $\tau = 0.51$.

Table 3: Classification performance of our shallow CNN architecture trained on *P-negGA* vs. Weimer et al. [10].

Surface no.	*Ours*		Weimer et al.	
	TPR	TNR	TPR	TNR
1	1.000	0.925	1.000	1.000
2	0.909	1.000	1.000	0.973
3	1.000	1.000	0.955	1.000
4	0.875	0.919	1.000	0.987
5	1.000	1.000	0.988	1.000
6	1.000	1.000	1.000	0.995
7	1.000	1.000	-	-
8	1.000	1.000	-	-
9	1.000	0.984	-	-
10	1.000	1.000	-	-

5 Conclusion

We find that our initial assumptions, which are: I.) We can use shallower CNN architectures in the domain of automated visual-inspection of surface anomalies, since the underlaying structures and patterns which need to be learned are limited; and II.) Since we are dealing with unbalanced sets, where each example provides positive and negative samples, learning can be done merely on positive examples; hold true in most cases. Considering our experimental setup, promising steps include scaling the ground truth to the interval of $[-1, 1]$ and performing minor data augmentation. Given the results in Table 3, merely Surface no. 4 needs to be further examined in order to determine the cause of the lower classification accuracy. It could be that in this case a deeper model would potentially perform better. In future work we plan to further improve the classification procedure by introducing a CNN architecture with a regression output in addition to the probability map output.

References

[1] Shuyue Chen, Jun Feng, and Ling 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.

[2] Shahrzad Faghih-Roohi, Siamak Hajizadeh, Alfredo Núñez, Robert Babuska, and Bart 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.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian 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.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[6] Jonathan Masci, Ueli Meier, Dan Ciresan, Jürgen Schmidhuber, and Gabriel 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.

[7] 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.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] Daniel Soukup and Reinhold 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.

[10] Daniel Weimer, Bernd Scholz-Reiter, and Moshe Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals-Manufacturing Technology*, 2016.

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