Steel Defect Classification with Max-Pooling Convolutional Neural Networks

Jonathan Masci, Ueli Meier, Dan Ciresan,
Jürgen Schmidhuber
IDSIA, USI and SUPSI
Galleria 2, 6928 Manno-Lugano,
Switzerland
{jonathan, ueli, dan, juergen}@idsia.ch

Gabriel Fricout
Arcelor Mittal
Maizières Research SA,
France
{gabriel.fricout@arcelormittal.com}

***Abstract*—We present a Max-Pooling Convolutional Neural Network approach for supervised steel defect classification. On a classification task with 7 defects, collected from a real production line, an error rate of 7% is obtained. Compared to SVM classifiers trained on commonly used feature descriptors our best net performs at least two times better. Not only we do obtain much better results, but the proposed method also works directly on raw pixel intensities of detected and segmented steel defects, avoiding further time consuming and hard to optimize ad-hoc preprocessing.**

I. Introduction

Machine vision based surface inspection technologies have gained a lot of interest from various industries to automate inspection systems, and to significantly improve overall product quality. A typical industry adopting these refined inspection tools is the rolled steel strip market. Real-time visual inspection of production lines is crucial to provide a product with ever fewer surface defects. Manual inspection, even though really accurate in case of few samples, is slow and prone to fatigue induced errors in high speed modern setups. This results in many additional costs which make this approach not only too expensive but also inapplicable. Automated machine vision inspection, one of the most investigated areas in the field of quality control, is fast and reliable achieving satisfactory results in many cases.

Combined developments in camera technologies, acquisition hardware and machine learning algorithms offer the required tools to meet speed, resolution and classification demands of current production lines. Even with the proper equipment and most advanced algorithms the problem of steel defect detection and classification remains non-trivial. Further improvements based on expert knowledge encoded in geometrical and shape-based features are difficult to achieve. A successful inspection algorithm should adaptively learn according to the changing data distribution, especially in modern dynamic processes where the production shifts from a product to another very quickly. On the other hand, all the expert knowledge and effort put in hand-crafted feature extraction is valuable and should complement any novel inspection algorithm.

A standard inspection system is coarsely divided in three main stages: image acquisition, feature extraction, and classification. The system is usually based on a set of hand-wired pipelines with partial or no self-adjustable parameters which makes the fine-tuning process of this industrial systems cumbersome, requiring much more human intervention than desired. In this work we focus on the two last pipeline stages and propose an approach based on Max-Pooling Convolutional Neural Networks (MPCNN) [1], [2], [3], [4], [5], that learn the features directly from labeled images using supervised learning. We show that the proposed method achieves state-of-the-art results on real world data and compare our approach to classifiers trained on classic feature descriptors.

There is not much literature about steel defect detection [6]. However, in a broader context the problem can be viewed as defect detection in textured material which has received considerable attention in computer vision [7], [8], [9]. In classical approaches, feature extraction is performed using the filter-bank paradigm, resulting in an architecture very similar to a MPCNN. Each image is convolved with a set of two-dimensional filters, whose structure and support come from prior knowledge about the task, and the result of the convolutions (filter responses) is later used by standard classifiers. A popular choice for the two-dimensional filters are Gabor-Wavelets that offer many interesting properties and have been successfully applied for defect detection in textured materials in general [10], for textile flaw detection [11] and face recognition [12] in particular. While being a very powerful technique, it has many drawbacks. First of all it is inherently a single layer architecture whereas deep multi-layer architectures are capable of extracting more powerful features [13]. Furthermore, the filter response vector after the first layer is high dimensional and requires further processing to be handled in real time/memory bounded systems.

The rest of the paper is organized as follows. We first review classical feature extraction methods and introduce the neural network framework. We then describe the data set and show results from the various experiments we performed.

II. Related work

When it comes to texture related descriptors the amount of available techniques is quite large, to say the least, making the selection process cumbersome and not easy to optimize for various products. Here we summarize the most notable descriptors which have been used in a similar context and compare them later on with our MPCNN approach.

- *Local Binary Patterns (LBP)* [14]: is an operator that focuses on the spatial structure of grey-level texture. For each pixel, the method takes a surrounding neighborhood and creates a binary descriptor which is intensity and rotation invariant based on the signs of differences between neighboring pixels.
- *Local Binary Pattern Histogram Fourier (LBP-HF)* [15]: is a rotation invariant descriptor computed from discrete Fourier transforms of LBP. The rotation invariance is computed on the histogram of non-invariant LBP, hence the rotation invariance is attained globally. The resulting features are invariant to rotations of the whole input signal but still retain information about the relative distribution of different orientations of uniform local binary patterns.
- *Monogenic-LBP* [16]: integrates the traditional Local Binary Pattern (LBP) operator with the other two rotation invariant measures, the local phase and the local surface type are computed by the 1st- and 2nd-order Riesz transforms.
- *Rotation invariant measure of local variance (VAR)* [17]: is a descriptor which is not gray-scale invariant as LBP but incorporates information about the contrast of local image texture. VAR in conjunction with LBP makes a very powerful rotation invariant descriptor of local image texture.
- *Histogram of Oriented Gradients (HOG)* [18]: is a method based on evaluating normalized local histograms of image gradient orientations in a dense grid. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions.
- *Pyramid of Histograms of Orientation Gradients (PHOG)* [19]: is an extension of the HOG descriptor that also considers the spatial locality of the descriptor's constituents.

III. MAX-POOLING CONVOLUTIONAL NEURAL NETWORKS (MPCNN)

Convolutional Neural Networks (CNN) are hierarchical models alternating two basic operations, convolution and subsampling, reminiscent of simple and complex cells in the primary visual cortex [20]. As CNN share weights, the number of free parameters does not grow proportionally with the input dimensions as in standard multi-layer networks. Therefore CNN scale well to real-sized images and excel in many object recognition benchmarks [5], [21], [22]. A CNN as depicted in Figure 1, consists of several basic building blocks briefly explained here:

- *Convolutional Layer*: performs a 2D filtering between input images x and a bank of filters w, producing another set of images h. A connection table CT indicates the input-output correspondences, filter responses from inputs connected to the same output image are linearly combined. Each row in CT is a connection and has the following semantic: (inputImage, filterId, outputImage). This layer performs following mapping

  $$h_j = \sum_{i,k \in CT_{i,k,j}} x_i * w_k \tag{1}$$

  where $*$ indicates the 2D valid convolution. Each filter w_k of a particular layer has the same size and defines, together with the size of the input, the size of the output images h_j. Then, a non-linear activation function (e.g. tanh, logistic, etc.) is applied to h just as for standard multi-layer networks.
- *Pooling Layer*: reduces the dimensionality of the input by a constant factor. The scope of this layer is not only to reduce the computational burden, but also to perform feature selection. The input images are tiled in non overlapping subregions from which only one output value is extracted. Common choices are maxima or average, usually shortened as Max-Pooling and Avg-Pooling. MaxPooling is generally favorable as it introduces small invariance to translation and distortion, leads to faster convergence and better generalization [23]. In this paper we will use only this kind of subsampling layer and hence the name MPCNN.
- *Fully Connected Layer*: this is the standard layer of a multi-layer network. Performs a linear combination of the input vector with a weight matrix. Either the network alternates convolutional and max-pooling layers such that at some stage a 1D feature vector is obtained (images of 1×1), or the resulting images are rearranged to have 1D shape. The output layer is always a fully connected layer with as many neurons as classes in the classification task. The outputs are normalized with a softmax activation function and therefore approximate posterior class probabilities.

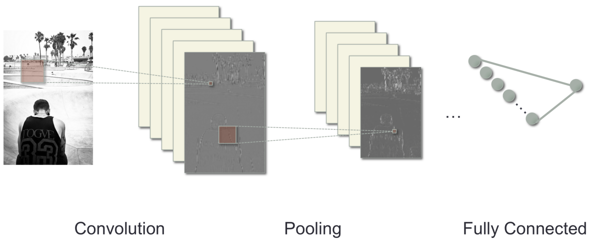

Fig. 1. A schematic representation of a Max-Pooling Convolutional Neural Network. Convolutional layers and max-pooling layers are stacked until the fully connected layers used for classification start.

A. Training Procedure

As for any supervised architecture the network is trained to predict the correct label for a given input pattern, minimizing the misclassification error over a set of labeled examples (i.e. the training set). Using the back-propagation algorithm [24] the gradient of the error function with respect to the adjustable parameters of the MPCNN is efficiently obtained, and any gradient based optimization algorithm can then be used. In general however, due to the highly non-linear nature of the error surface, a stochastic gradient descent procedure is preferred as it usually avoids being stuck in poor local minima. For all experiments we anneal the learning rate and update the weights after each sample. A learning epoch is complete when all the samples of the training set have been visited once. Note that random permutation of the samples prior to each epoch is very important to obtain i.i.d patterns.

The error back-propagation for a convolutional layer is given by

$$\delta_i = \sum_{k,j \in CT_{i,k,j}} \delta_j * w_k \tag{2}$$

where $*$ indicates the 2D full correlation, equivalent to a convolution with a kernel flipped along both axes. The gradient is computed as

$$\nabla w_k = x_i * \delta_j \tag{3}$$

where in this case $*$ indicates the 2D valid convolution. In equation 3 there is no summation as each kernel is only used once, there are no multiple images sharing the same kernel.

IV. Dataset

The experiments are performed on data collected on an actual production line. The dataset is composed of a subset of 7 defects, chosen because of their intra-class variabilities which makes learning difficult. In figure 2 two instances of the same defect are shown to illustrate the intraclass variability of this data. The images come in big patches with the detected region of interest that the segmentation stage produces. This stage can obviously miss the defect and create false alarms. In this study we do not consider errors from the detection/segmentation pipeline and consider each segmented region of interest as a correctly labeled sample. This implies that in addition to high intraclass variability the classification algorithm also has to deal with false positives in the training set.

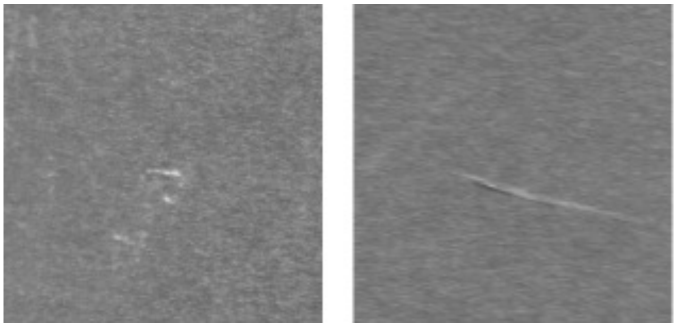

Fig. 2. Two instances of the same defect class.

Contrary to classical machine vision approaches where there are no constraints on the input dimensions for the feature extraction stage, MPCNNs need a constant sized input as they perform feature extraction and classification jointly. We therefore resize all the defects, preserving the aspect ratio and minimizing the overall down/up-sampling rate according to the distribution of image dimension over the training set (figure 3).

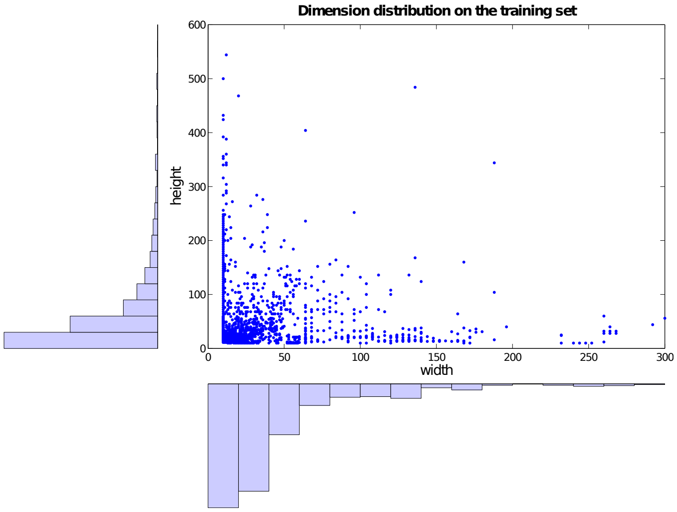

Fig. 3. Each point corresponds to the width and height of an image from the training set, histograms of the width and height distribution are also shown.

We decide to resize the images to 150×150 pixels, padding with surrounding pixels in the smaller patch whenever necessary. If the region of interest is smaller than the target size we take surrounding pixels with no rescaling; otherwise we downsample the bigger patches. In figure 4 a sample from each of the 7 defects detected on the superior (first row) as well as on the inferior (second row) part of the steel strip is shown. There is no correspondence between defects from the superior and inferior part, and no additional information can be extracted as each image is considered as an independent sample of a particular defect. In total the training set consists of 2281 images and the test set consists of 646 images.

In classical approaches, a histogram of the features is created in order to obtain a constant sized feature vector used for classification. For all experiments using classical feature extraction techniques we adopt this approach. We keep the original images avoiding artifacts that might ruin the quality of the features. For example, if a defect covered only 10% of the image and we zero-padded, the resulting histogram would almost be flat and the actual information regarding the defect would be lost.

V. Results

A. Standard Features

For each of the standard features we tested, the code provided by the respective authors was used, so a margin of improvement might be achieved by fine tuning the parameters. For LBP we combine rotation invariant and non rotation invariant features by simple concatenation the two feature vectors. As always, prior knowledge and experience are important, but fine tuning the parameters of the feature extraction is usually harder than for MPCNNs, especially for LBP which depends on many parameters and is available in many variants.

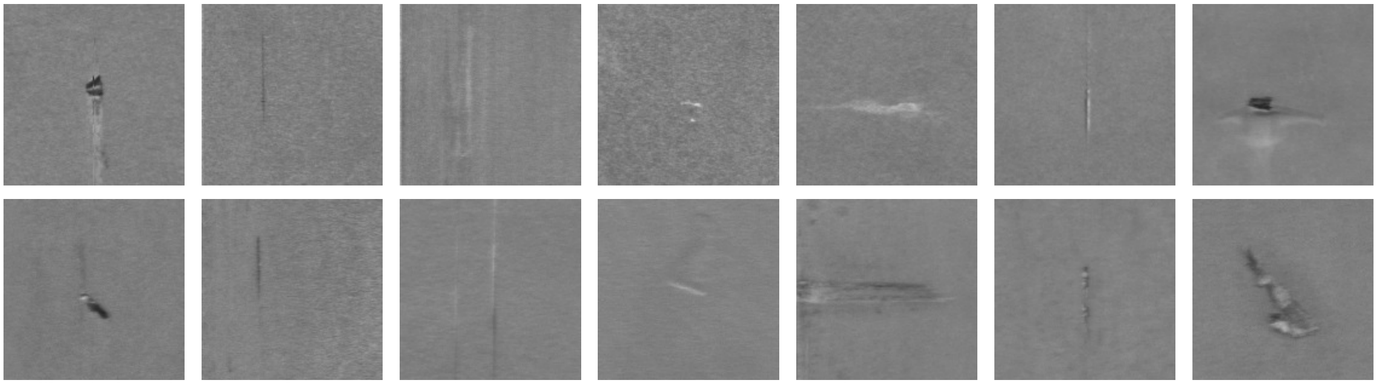

Fig. 4. A sample from each of the seven defects in the dataset. First row: superior part of the strip; second row: inferior part of the strip. There is no correspondence between the two images for a given instance of the defect.

The extracted features, which are histograms, are locally normalized between 0 and 1 and then fed to a classifier. We opt for a multi-layer perceptron (MLP) just as for the MPCNN, where the last to layers are an MLP. It is also easier to extract values that can afterwards be interpreted as posterior probabilities (see section V-C). We used a single layer architecture with 100 hidden nodes [1], number of inputs given by the dimensionality of a given feature representation and tanh activation function; training is performed for a total of 300 epochs. The output layer has 7 neurons, one for each defect class, that are normalized with a softmax activation function. We also tested a SVM with RBF-kernel, whose kernel parameter γ and penalty term C are optimized over following grid: $\gamma = [2^{-15}, 2^{-13}, ..., 2^3]$ and $C = [2^{-5}, 2^{-3}, ..., 2^{15}]$ by a 5-fold cross-validation. Results for both classifiers are shown in Table I.

TABLE I
CLASSIFICATION ERRORS FOR A MLP/SVM TRAINED ON THE HISTOGRAMS GENERATED BY CLASSICAL FEATURES.

	MLP		**SVM**	
Feature (#Dims)	train	test	train	test
LBP (274)	28.54	32.28	5.62	26.79
LBP-HF (76)	29.24	40.09	8.82	29.11
MONO-LBP (540)	7.40	21.20	4.61	19.66
VAR (10000)	27.92	43.34	21.71	42.88
HOG (81)	7.89	21.36	0.84	19.35
PHOG (680)	0.04	19.04	0.05	15.48

B. MPCNN

We train two different architectures with stochastic gradient descent and annealed learning rate. We perform experiments with and without random translation of max. ±15% of the image size. Since no padding region is used, border effects arise. In order to minimize these we opt for the simplest solution, and assign to each unknown pixel the intensity of the closest pixel with respect to the euclidean distance. This may introduce problems when a defect is close to the border as it might possibly disappear due to the translation factor. Training stops when either the validation error becomes 0, the learning rate reaches its predefined minimum or there is no improvement on the validation set for 50 consecutive epochs. The undistorted, original training set is used as validation set to cope with the limited amount of valuable labeled training data. A GPU implementation of a MPCNN is used to speed up training, on a GeForce GTX 460 one training epoch takes about 90s, a speed-up of 20-40 (depending on the network topology) with respect to an optimized CPU implementation.

The first architecture has 5 hidden layers, a convolutional layer with 50 maps and filters of size 19×19, a max-pooling layer of size 4×4, a convolutional layer with 100 maps and filters of size 13×13, a max-pooling layer of size 3×3, a fully connected layer with 100 neurons, a fully connected layer with 7 output classes (5HL-MPCNN). The second architecture has 7 hidden layers, a convolutional layer with 50 maps and filters of size 11×11, a max-pooling layer of size 4×4, a convolutional layer with 100 maps and filters of size 6×6, a max-pooling layer of size 3×3, a convolutional layer with 150 maps and filters of size 5×5, a max-pooling layer of size 3×3, a fully connected layer with 100 neurons, a fully connected layer with 7 output classes (7HL-MPCNN).

Classification error for training and test set are shown for both architectures trained for 50 epochs with and without translation. All layers of the nets in Tab. II have been trained whereas the first layer for the nets in Tab. III was kept fixed during the learning process. That is the first convolutional layer performs random filtering, reducing number of free parameters and hence training time without degrading classification performance. As a matter of fact it even improves generalization when no translations are used [25]. Each experiment is repeated five times with different random initializations, since any iterative gradient based optimization technique depends on the starting model. The deeper net yields better results and translating images prior to training by a maximal amount of 15% further increases performance. The additional translations serve as a regularizer and improve generalization capabilities of the trained nets to the unseen test data (i.e. a translated

[1] We also tested MLPs with more hidden units and an additional hidden layer with no considerable improvement.

image of a particular defect, still belongs the same defect) especially when the amount of training data is scarce. All the MPCNN yield lower error rates than any of the feature based classifiers (Tab. I). The best 7HL-MPCNN with an error rate of 6.97% with and 8.20% without translations clearly outperforms the best feature based classifier, PHOG with an error of 15.48%. This illustrates the power and potential of the proposed architecture, for defect classification in textured materials.

TABLE II
CLASSIFICATION RESULTS FOR TWO DIFFERENT MPCNN ARCHITECTURES TRAINED FOR 5 INDEPENDENT INITIALIZATIONS (RUN1-5) WITH AND WITHOUT 15% TRANSLATION, ALL LAYERS ARE TRAINED.

	5HL-MPCNN		7HL-MPCNN	
	No Trans.	15% Trans.	No Trans.	15% Trans.
run1	14.86	7.74	13.62	7.43
run2	14.09	10.84	12.38	**6.81**
run3	15.79	11.15	**10.99**	8.05
run4	**13.78**	9.75	11.15	10.06
run5	15.64	**9.60**	13.78	8.20

TABLE III
CLASSIFICATION RESULTS FOR TWO DIFFERENT MPCNN ARCHITECTURES TRAINED FOR 5 INDEPENDENT INITIALIZATIONS (RUN1-5) WITH AND WITHOUT 15% TRANSLATION, FIRST CONVOLUTIONAL LAYER IS NOT TRAINED.

	5HL-MPCNN		7HL-MPCNN	
	No Trans.	15% Trans.	No Trans.	15% Trans.
run1	15.02	11.30	9.91	7.28
run2	**12.07**	13.16	8.98	7.28
run3	13.31	11.92	**8.20**	**6.97**
run4	14.55	12.54	9.29	8.05
run5	13.00	**10.99**	8.67	8.20

In Figure 5-left the confusion matrix of the best MPCNN from Tab. III is shown, where the rows represent the true classes and the columns the predicted classes. On the diagonal the per class percentage of correctly classified samples is shown, all off-diagonal entries in each row correspond to the wrongly classified samples for a particular class. For example, 14% (first entry in row 6) of samples from defect class 6 are wrongly classified as class 0.

TABLE IV
DETAILED NETWORKS STRUCTURE. THE TIME PER SAMPLE REFERS TO THE TIME REQUIRED FOR A TRAINED NETWORK TO PRODUCE THE CLASS PREDICTION.

Network	#parameters	#connections	time per sample
5HL-MPCNN	1.35M	688M	11.3ms
7HL-MPCNN	622k	295M	6.2ms

C. *Committee of classifiers*

Combining the output of several classifiers is an easy and effective way of boosting the performance. If the errors of different classifiers have zero mean and are uncorrelated with each other, then the average error might be reduced by a factor of M simply by averaging the output of the M models [26]. In practice, error of models trained on similar data tends to be highly correlated. To avoid this problem predictions of various classifiers trained on differently normalized data can be combined [27]. Along similar lines the same classifier can be trained on random subsets of the training set (bootstrap aggregation technique [28]), or different types of classifiers can be trained on the same data [29]. Here we combine classifiers trained on different features, harnessing the complementary information content of the various feature descriptors.

In Table V the results of the three best committees out of all possible committees with at least 2 out of the 6 classifiers trained on the 6 different feature descriptors. The best committee decreased the error rate by 5% with respect to the best single classifier. Note, however, that even the three best committees have a much bigger error rate compared to the MPCNN (Tabs. II, III). In Figure 5 we clearly see that using a committee of classifiers considerably boosts the recognition rate (compare middle and right matrices). We can also see that a simple MPCNN performs always better in the per-class evaluation (diagonal values) except for defect number 2 where a committee reaches almost perfect accuracy.

TABLE V
CLASSIFICATION RESULTS FOR THE TOP-3 COMMITTEES OF MLPS TRAINED ON THE HISTOGRAM GENERATED BY CLASSICAL FEATURES. WE TAKE THE BEST COMBINATION EVALUATED ON THE TRAINING SET. NOTE THE CONSIDERABLE IMPROVEMENT OVER A CONVENTIONAL SINGLE CLASSIFIER APPROACH (SEE TABLE I)

Best Combination	train	test
HOG, PHOG, LBP-HF	0.04	11.45
LBP, HOG, PHOG	0.04	13.46
PHOG, MONO-LBP	0.04	10.99
ALL FEATURES	2.84	11.60

VI. CONCLUSIONS

We presented a steel defect classification approach based on Max-Pooling Convolutional Neural Networks which is able to perform supervised feature extraction directly from the pixel representation of the steel defect images. We showed that without any prior knowledge excellent results are achieved, outperforming any classifier trained on feature descriptors commonly used for defect detection in textured materials. The best MPCNN with an error rate of 7% clearly outperforms the best classifier trained on PHOG features with an error rate of 15%. We conclude that for defect classification in textured materials, the proposed method is a viable alternative to standard feature descriptors.

It also scales well to multivariate images (i.e. color, hyperspectral) where the input will map to more than one channel as in gray-scale. This is a great advantage over hand-crafted features which are hard to be extended to such domains where even prior knowledge is still not well consolidated.

ACKNOWLEDGMENTS

This work was financed by ArcelorMittal.

	0	1	2	3	4	5	6
0	0.91	0.01	0.01				0.07
1	0.09	0.85				0.07	
2	0.01		0.99				
3	0.04			0.94	0.02		
4	0.02			0.02	0.97		
5		0.05	0.11			0.84	
6	0.14						0.86

	0	1	2	3	4	5	6
0	0.80	0.01	0.01	0.08	0.02	0.01	0.08
1	0.11	0.89					
2	0.03	0.02	0.94			0.01	
3	0.14		0.02	0.83	0.01		
4				0.09	0.91		
5	0.05	0.19				0.76	
6	0.25				0.01		0.74

	0	1	2	3	4	5	6
0	0.85	0.01		0.03	0.01	0.01	0.11
1	0.12	0.88					
2	0.01		0.98			0.01	
3	0.09		0.01	0.88	0.02	0.01	
4			0.02		0.98		
5	0.05	0.05		0.14		0.76	
6	0.20						0.80

Fig. 5. Confusion matrices for the best classifiers. Left: MPCNN, middle: PHOG, right: PHOG + MONO-LBP committee. Only on defect number 2 the classical features obtained a better result than our MPCNN. Also note the non marginal improvement of a committee w.r.t. the single best classifier.

References

[1] K. Fukushima, "Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[3] S. Behnke, *Hierarchical Neural Networks for Image Interpretation*, ser. Lecture Notes in Computer Science. Springer, 2003, vol. 2766.

[4] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition*, 2003, pp. 958–963.

[5] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *International Joint Conference on Artificial Intelligence*, 2011, pp. 1237–1242.

[6] L. Martins, F. Pá anddua, and P. Almeida, "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*, 2010, pp. 1081 –1086.

[7] T. K. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[8] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 691–698.

[9] ——, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[10] A. Kumar and G. K. Pang, "Defect detection in textured materials using gabor filters," *IEEE Transactions on Industry Application*, vol. 38, no. 2, pp. 425–440, 2002.

[11] A. Bodnarova, M. Bennamoun, and S. Latham, "Optimal gabor filters for textile flaw detection," *Pattern Recognition*, vol. 35, no. 35, pp. 2973–2991, 2002.

[12] O. Ayinde and Y.-H. Yang, "Face recognition approach based on rank correlation of gabor-filtered images," *Pattern Recognition*, vol. 35, no. 35, pp. 1275–1289, 2002.

[13] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. International Conference on Computer Vision (ICCV'09)*. IEEE, 2009.

[14] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[15] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Proceedings of the 16th Scandinavian Conference on Image Analysis*, ser. SCIA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 61–70.

[16] L. Zhang, L. Z. 0006, Z. Guo, and D. Zhang, "Monogenic-lbp: A new approach for rotation invariant texture classification," in *ICIP*. IEEE, 2010, pp. 2677–2680.

[17] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971 –987, Jul. 2002.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893.

[19] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, ser. CIVR '07. New York, NY, USA: ACM, 2007, pp. 401–408.

[20] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex." *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, March 1968.

[21] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *International Joint Conference on Neural Networks*, 2011, pp. 1918–1921.

[22] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *International Conference on Document Analysis and Recognition*, 2011, pp. 1250–1254.

[23] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *International Conference on Artificial Neural Networks*, 2010.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. [Online]. Available: http://portal.acm.org/citation.cfm?id=104279.104293

[25] A. M. Saxe, P. W. Koh, Z. Chen, M. Bh, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning," in *International Conference on Machine Learning*, 2011.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[27] U. Meier, D. C. Ciresan, L. M. Gambardella, and J. Schmidhuber, "Better digit recognition with a committee of simple neural nets," in *International Conference on Document Analysis and Recognition*, 2011, pp. 1135–1139.

[28] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[29] E. Grosicki and H. E. Abed, "Icdar 2011 - french handwriting recognition competition," in *11th International Conference on Document Analysis and Recognition*, 2011, pp. 1459–1463.