Training of Classifiers Using Virtual Samples Only

Annika Kuhl* Lars Krüger Christian Wöhler Ulrich Kreßel
DaimlerChrysler AG Research and Technology
Wilhelm-Runge-Str. 11, D-89081 Ulm, Germany
{annika.kuhl, lars.krueger, christian.woehler, ulrich.kressel}@daimlerchrysler.com

Abstract

This paper describes the training of classifiers entirely based on virtual images, rendered by a ray-tracing software. Two classifiers, a support vector machine and a polynomial classifier, are trained solely with virtual samples and used for classification of real samples. The objects to be distiguished are holes vs. garbage (non-holes) out of a set of hole candidates in images of flanges. We analysed the effect of different classifier parameters and manipulation of the virtual samples. Error rates of 1.6 % on real test samples are achieved.

1 Introduction

In order to obtain a robust classifier the training set needs to contain samples representing the problem adequately. In case of image classification this means a sufficiently large variation in spatial position, illumination, and/or coloring.

Unfortunately this amount of images is not available for many applications or there exists an imbalance between the number of samples per class. Both leads to underrepresentation yielding poor classification results due to overfitting. This problem can be solved by using virtual images for the training. Modern photo-realistic rendering techniques make it possible to generate these images from a CAD model of the object. Using these data to render the scene allows for modification of spacial position, color and/or illumination. With a sufficient number of these appearance modifications a sufficient amount of different training images are obtained and can be used to train a robust classifier.

The traditional way to cope with the small training set problem is interpolation, also known as noise injection. Virtual samples are generated by interpolating real samples as done by training ALVINN [10]. In [2] gaussian noise is added to the input in order to generate a sufficient number of training samples. Abu-Mostafa [1] solves the problem by generating virtual samples out of the domain knowledge of the problem, called hints. In [6] an existing set of real training samples is extended with generated virtual samples in order to improve representational capability. All these methods try to improve generalization by enlarging the training set consisting of real data with virtual samples as analysed by [4] and [12].

All these noise injection techniques require an already existent training set which is extended to obtain a more robust classifier. The weak point is that they interpolate in sample space. Since the imaging process can be seen as a function whos inverse function can not obtained, there is no connection between a sample of e.g. reduced brightness and the exact enery settings of the lamps. This makes it either factually or economically impossible to obtain a sufficiently large training set, especially before the actual image processing station is already built.

On the other hand modern factory planning makes it possible to design a future factory down to very small details. It is desirable to include even the image processing algorithms in this planning process, so that the number of cameras, lights etc. is known before the actual fabrication or even the construction of the factory itself starts. It is clear that such planned quality control stations – if they include classifiers – should be trained before the actual processing starts. The virtual imaging processing process (a.k.a rendering) is non-invertable too, but it is possible to automatically try a lot of combination of e.g. lighting positions even before the factory construction starts. In this paper we will therefore describe how to train classifiers with virtual samples only, and report the resulting effect on generalization performance and error rate.

*This project was done during an internship within the study of computer science at the Technical University of Ilmenau, P.O. Box 100565, D-98684 Ilmenau, Germany. annika.kuhl@stud.tu-ilmenau.de

2 System Overview

The example used throughout this paper is a flange mounted on a gearbox (see fig. 1). The task of quality inspection is to assure the correctness of the type of the mounted flange. A type verification on these flanges can be done using the number and positions of the flange's holes. Using a simple detector hole candidates are generated which are to be separated by a classifier into holes and garbage.

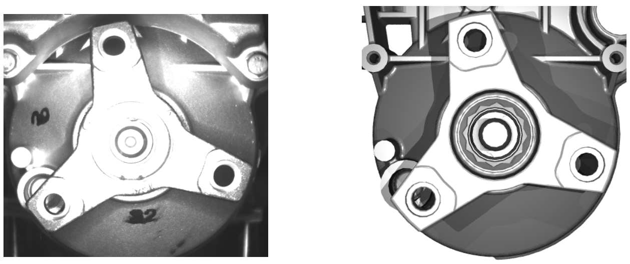

Figure 1. Left: Real image. Right: Virtual image.

For hole detection a binary image is created using an adaptive threshold. The binary connected components algorithm [8] is used to search black regions enclosed in white regions which form the hole candidates. These candidates are normalized to 16 by 16 pixel in size.

For classification a standard support vector machine [13, 7] with polynomial kernel was implemented. The SVM attempts to separate objects belonging to two given classes in n-dimensional real spaces by a polynomial curve. This dividing line is defined by a kernel function wherein MINOS [9] is used as the solver for the quadratic optimization problem.

In order to decrease the error rate suitable values for the polynomial degree and the cost value are chosen.

For comparison a polynomial classifier [11] is trained and tested on the same samples as the SVM. To reduce the feature space a principal component analysis [5] is applied in the case of the polynomial classifier.

3 Data Set

For photo-realistic images a sufficiently exact description of the real scene and adequate rendering software is required. We use the open source ray-tracer POVRay [3] for its simplicity and power. The scene description was derived from the CAD model of the object using various export filters and converters. Though the CAD model contains no information about material properties, they are adjusted to achieve a realistic impression of the rendered object.

Using the aforementioned preprocessing steps 8824 virtual samples and 887 real samples were obtained. The sets contain approximately the same number of hole and garbage samples. For some examples see fig. 2.

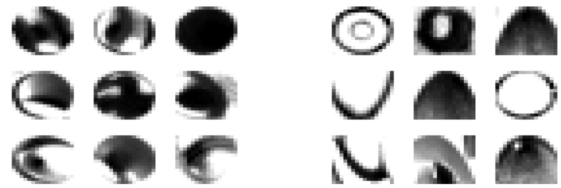

Figure 2. An assortment of real holes (left) and garbage (right).

The appearance modifications during the rendering process included the position of the object, the overall illumination of the scene and the surface gray level. Especially the rotation of the flange is very important. For some orientations the flange partially occludes a gearbox mounting point as seen in fig. 1. This changes the appearance of the hole while leaving the elevated ring around the hole unchanged. Obviously this has to be represented in the data sets.

Figure 3. Some support vectors of class 1 - virtual holes (left) and class 2 - virtual garbage (right)

We vary the surface gray level from dark (0.3) to light (0.9) to cover variations in the real images as seen in fig. 4. Additional blurring (fig .8) increases the similarity to the real images. This blurring accounts for the point spread function of the camera lens.

4 Classification Results

The linear support vector machine trained on a reduced sample set achieves error rates of about 5 %. This reduced sample set contains no variation in position or illumination with 2195 samples of each class only. The complete sample set consisting of 8824 samples covers the whole variation in position of the flange and some variation in illumination and surface gray

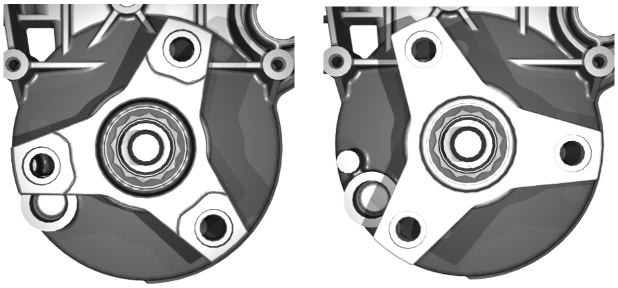

Figure 4. Surface gray level of 0.3 (left) and 0.9 (right)

level. With this training set the error rate drops to 3.5 %. By increasing the sample set to a greater variation in illumination, surface gray level or focus no significant improvement is achieved. A variation of the polynomial degree of the SVM finally leads to an error rate of 1.6 % with the polynomial degree set to 4. This is achieved by 7 wrong classified hole samples out of 388 and 7 wrong classified garbage samples out of 499. Figure 3 shows some support vectors for this run. Different cost values have no effect on error rates due to the linear separability of the training set.

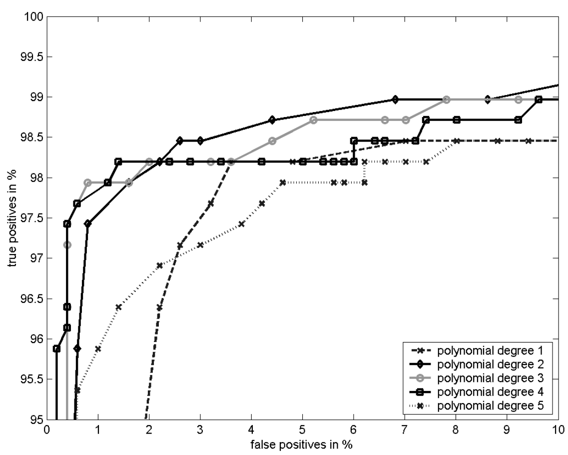

Figure 5. ROC curves on the test set of the support vector machine for different polynomial degrees

The ROC curve in fig. 5 shows false positives against true positives, whereas true positives are the percentage of correctly classified holes and false positives are the percentage of wrong classified garbage. As seen in the figure a polynomial degree of 2 is sufficient to solve the classification problem. The classification rate thereby amount to 98.2 %. This decreases the error by 0.9 % in comparison to a linear SVM. Increasing the polynomial degree to values of 5 and higher increases the error rate due to overfitting.

In order to validate these results a classical linear approach is chosen. Therefore a polynomial classifier is trained and tested on the same data. A polynomial degree of 1 and a features space of 256 results in non-satisfying error rates. This behaviour is ascribed to overfitting. To reduce the feature space a principle component analysis is accomplished. A reduction to 50 features, corresponding to a reconstruction rate of 99.22 %, increases the classification rate to about 95 %.

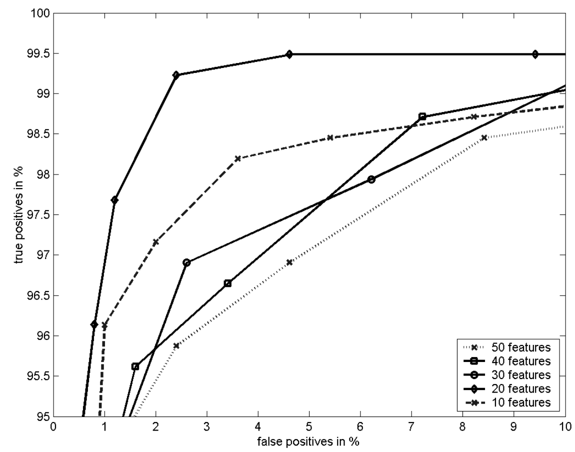

Figure 6. Linear polynomial classifier for several numbers of principle components used for classification

The ROC curve in fig. 6 shows that down to 20 principle components the error rate decreases continuously. For 20 principle components the classification rate amounts to 97.5 %. A reduction to less than 20 features leads to an increasing reconstruction error and therefore to increasing classification error rates. We conclude that the information is contained mainly within the first 20 principle components. This hypothesis is confirmed by fig. 7. It shows the first and second principle component of the real samples. The distribution of the pattern decomposition already appears as two almost distinct clusters.

The results achieved by the linear polynomial classifier without reduction in feature space are much worse than the SVM with a linear kernel. This is due to the regularisation ability of the SVM, whereas a polynomial classifier tends to suffer from overfitting in a high-dimensional feature space, though a lower-dimensional feature space implies a higher reconstruction error. In this example linear polynomial classifier using the first 20 principle components achieves similar rates as the second-order SVM trained on the same data set without dimensional reduction.

5 Summary and Conclusion

In this paper we have shown the results of support vector machines and polynomial classifiers trained on virtual samples only. We analysed the effect of different classifier parameters and manipulation of virtual samples. To render these virtual images a sufficiently exact description of the real object is required. The difficulty is to vary rendering parameters such as illumination or surface gray level to achieve a realistic appearance.

We have shown that a classifier trained only on virtual samples achieves good classification rates on real test data. Concerning generalisation, the SVM is superior to the polynomial classifier due to its regularisation ability, because no principle component analysis is needed to reduce the input dimension.

The main advantage of rendering virtual images is the possibility to rapidly generating a large amount of different samples which is necessary to obtain a robust classifier. For many applications this high amount of samples is expensive or impossible to obtain, such that rendering by virtual techniques is an economical and successful alternative.

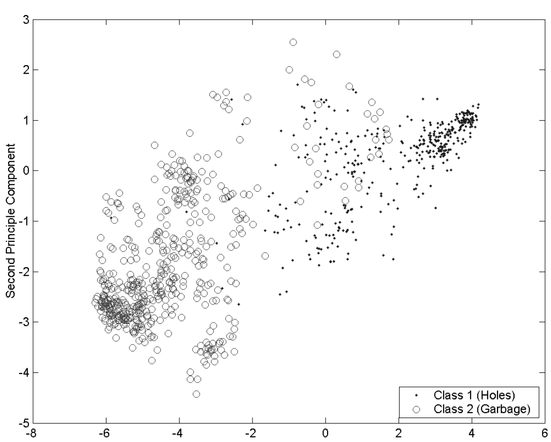

Figure 7. First and second principle components of the real test samples

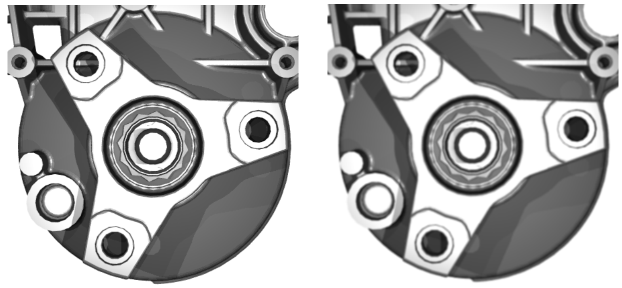

Figure 8. Virtual image without (left) and with blurring (right)

References

[1] A. Abu-Mostafa. Hints. *Neural Computation*, 7:639–671, 1995.

[2] G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, 7:613–674, 1996.

[3] D. Buck and A. Collins. Povray homepage (http://www.povray.org).

[4] S. Cho and K. Cha. Evolution of neural network training set through addition of virtual samples. *International Conference on Evolutionary Computation*, pages 685–688, 1996.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification, second edition. *John Wiley and Sons. Inc.*, New York, 2001.

[6] D. M. Gavrila and J. Giebel. Virtual sample generation for template-based shape matching. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1:676 – 681, Kauai, U.S.A., 2001.

[7] U. Kressel and J. Schürmann. Pattern classification techniques based on function approximation. *Handbook on Optical Character Recognition and Document Image Analysis, Chapter 2*, pages 49 – 78, World Scientific Publishing Company, 1997.

[8] E. Mandler and M. Oberländer. One pass encoding of connected components in multi-valued images. *IEEE Int. Conf. on Pattern Recognition*, pages 64 – 69, Atlantic City, 1990.

[9] B. Murtagh and M. Saunders. Largescale linearly constrained optimization. *Mathematical Programming*, 14:41–72, 1978.

[10] D. Pomerleau. Neural network perception for mobile robot guidance. *Kluwer Academic Publishing*, 1993.

[11] J. Schürmann. Pattern classification. *Wiley - Interscience*, New York, 1996.

[12] M. Skurichina, S. Raudys, and R. P. W. Duin. K-nearest neighbours directed noise injection in multilayer perceptron training.

[13] V. N. Vapnik. The nature of statistical learning theory. *Springer, New York*, 1995.