SAR ATR by a Combination of Convolutional Neural Network and Support Vector Machines

SIMON A. WAGNER
Fraunhofer Institute for High Frequency Physics and Radar Techniques
Wachtberg, Germany

A combination of a convolutional neural network, which belongs to the deep learning research field, and support vector machines is presented as an efficient automatic target recognition system. Additional training methods that incorporate prior knowledge to the classifier and further improve its robustness against imaging errors and target variations are also presented. These methods generate artificial training data by elastic distortion and affine transformations that represent typical examples of image errors, like a changing range scale dependent on the depression angle or an incorrectly estimated aspect angle. With these examples presented to the classifier during the training, the system should become invariant against these variations and thus more robust. For the classification, the spotlight synthetic aperture radar images of the moving and stationary target acquisition and recognition database are used. Results are shown for the ten class database with a forced decision classification as well as with rejection class.

Manuscript received January 13, 2016; revised May 30, 2016; released for publication July 11, 2016.

DOI. No. 10.1109/TAES.2016.160061.

Refereeing of this contribution was handled by A. Charlish.

Authors' address: S. A. Wagner, Fraunhofer Institute for High Frequency Physics and Radar Techniques, FHR, Cognitive Radar, z. Hd. Simon Wagner, Fraunhoferstraße 20, Wachtberg 53343, Germany, E-mail: (simon.wagner@fhr.fraunhofer.de).

I. INTRODUCTION

Over the last years several different systems for automatic target recognition (ATR) have been introduced [1–4], wherein the used data and techniques depend heavily on the scenario. For ground targets, mainly synthetic aperture radar (SAR) images are used for classification [3, 5], whereas for air targets research is mainly focused on high range resolution profiles [2, 6, 7] and the micro Doppler effect [8, 9], especially jet engine modulation [4, 10]. Also, inverse SAR (ISAR) images have been considered for the classification of air targets [1]. The proposed classifier is also applicable to ISAR images, but this work is focused on the identification of ground targets in SAR images. This scenario is based on the assumption that the target was successfully detected.

Currently a new radar target recognition approach based on deep learning and convolutional neural networks (CNNs) is becoming more established [11–16]. But to the knowledge of the author the first application of CNNs and thus deep learning applied to target recognition in SAR images was published in [17]. The work presented in this paper is an extension of [17] with training methods that increase the robustness against imaging errors in radar imaging. Some of these training methods and their benefit to ATR systems were already presented in [18].

In other fields of image based classification, like optical character recognition, CNNs are very well established and are the state of the art classification systems [19]. CNNs are a part of the deep learning field, but with the advantage that they can be trained by standard backpropagation algorithms. The field of deep learning in general relies on computational power and became popular with the use of graphical processor units for computation and the possibility to initialize the networks via a layerwise pretraining [20]. The structure of the used CNN and the whole classification system is presented in Section III. Since it is often required that a rejection class is included to reduce the probability of false alarms and to reject unknown objects in ATR applications, the presented classifier is implemented in two ways: with and without rejection class.

One idea to overcome the problem of limited training data, which all supervised learning machines have in common, is to generate virtual examples out of known data [21]. This idea was applied to SAR data in [22] by presenting negative examples, which were generated by permutations of known real SAR images. For this permutation, the images are divided into a fixed number of small patches and these patches are randomly permuted to generate images that show no clearly visible target. With these artificial negative examples the rejection of unknown targets was increased. In the field of optical character recognition this method was used with positive examples to build classifiers with an invariance against expected imaging errors or handwriting variations [23, 24]. In this work, we also use the way of presenting positive examples that represent typical imaging effects in SAR imaging. By

TABLE I
Number of Images Per Class

	Training	Test
Class 1 (BMP-2)	698	587
Class 2 (BTR-70)	233	196
Class 3 (T-72)	691	582
Class 4 (BTR-60)	256	195
Class 5 (2S1)	299	274
Class 6 (BRDM-2)	298	274
Class 7 (D7)	299	274
Class 8 (T-62)	299	273
Class 9 (ZIL-131)	299	274
Class 10 (ZSU-23/4)	299	274
Complete	3671	3203

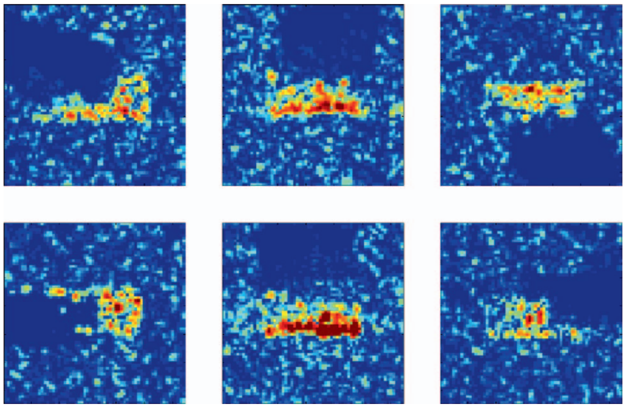


Fig. 1. Rotated and cropped images.

presenting these artificial images an increased robustness against imaging errors should be achieved. A detailed description of this procedure and the different transformations are given in Section IV. A successful application of this technique to high range resolution profiles was shown in [25], where the artificial data was not generated by a transformation, but as realizations from the estimated probability distribution function.

As already mentioned, this paper is focused on the identification of ground targets and as a database we use SAR images of the moving and stationary target acquisition and recognition (MSTAR) data [26]. A detailed description of the data is given in Section II and classification results are shown in Section V.

II. DATA DESCRIPTION AND PREPROCESSING

The used MSTAR dataset consists of X-band spotlight-SAR [27] images with 0.3 m resolution in each direction. The targets have been measured over the full 360° azimuth angles with 1°—5° increments. The target data is presented as subimage chips centered on the target with a size of 128 × 128 pixels. Data collected at 15° depression angle is used for testing (3203 images) and data with 17° depression angle for training (3671 images). The distribution of the images over the different targets is given in Table I.

Before the training and the classification can be done, some preprocessing of the data is required. To compare the images, all of them are rotated by their specific aspect angle. After that, all images show the targets at the same orientation. For this task, the aspect angle included in the data set was used. In one of our experiments the given angles are disturbed by a noise vector to get a more realistic scenario, since the estimation of the aspect angle is a difficult problem [28]. But also with the same orientation of the target there are still differences in the images, since with the aspect angle the illumination and the SAR-shadow differs.

After the rotation, areas with zeros occur at the corners of the images. To remove these and to reduce the dimension of the input space only the center area of size 64 × 64 pixels is kept for training and classification. This area is still large enough that the complete target fits into it and the dimension is computationally more manageable. A further benefit of the data cropping is that a lot of clutter is removed, since it was shown in [3] that the clutter of the MSTAR data shows such a high correlation between the training and the test images that a classification with only the clutter content of the images is possible.

As the last step, to get real-valued input images with an appropriate dynamic range, all images are converted to a logarithmic scale. Some examples that show the variability of class one with a changing aspect angle are depicted in Fig. 1, in which all images are in the same dynamic range.

III. CLASSIFICATION SYSTEM

The proposed classifier is a combination of a CNN and support vector machines (SVMs). In this section the two different parts are described independently and in the last part the combination of both systems is presented. A more detailed description of CNNs can be found in the original papers [19, 29] and further details about SVMs in textbooks like [30, 31].

A. Convolutional Neural Networks

CNN's are a special form of multilayer perceptrons, which are designed specifically to recognize two-dimensional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion [30]. This kind of network is inspired by the visual system of the brain and is part of the deep learning research field. For many years, CNNs were the only type of deep neural network that could efficiently be trained due to its structure using the technique of weight sharing [32]. The basic structure of the network used in this paper is shown in Fig. 2.

The first three layers of this network are an alternation of convolutional and subsampling layers. With these layers a feature extraction stage is created, where each neuron in the first convolutional layer takes its inputs from a local receptive field in the input image. The neurons of the first layer are separated into feature maps, each of them corresponding to one local spatial feature that is

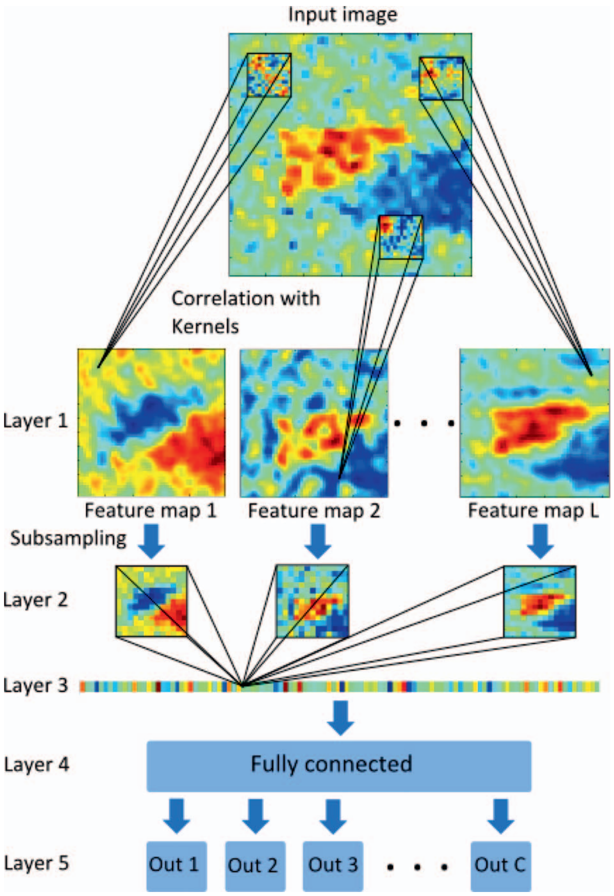


Fig. 2. Structure of the used convolutional neural network.

learned during the training. Because all neurons in one feature map use the same weights, and the local receptive field of neighboring neurons is shifted only by one pixel in one direction, the values of each feature map correspond to the result of the two-dimensional correlation of the input image with the learned kernel of each particular feature map.

In the input image of Fig. 2 one target is visible in the center of the image. The correlation with the different kernels is visualized for three examples. The learned kernels are depicted inside the black squares on the input image and the result of the correlation can be seen in the feature maps of the first layer.

The second layer of the network is a subsampling layer and performs a reduction of the dimension by a factor of four. With this reduction, the exact position of the feature becomes less important and it reduces the sensitivity to other forms of distortion [30]. The subsampling is done by averaging an area of 4 × 4 pixels, multiplying it with a weight w_j and adding a trainable bias b_j.

The third layer is a convolutional layer again and relates the features found in the image to each other. This layer is trained to find patterns of features, which can be separated by the subsequent layers and discriminate the different classes.

The last two layers of the network form the decision part of the system and are fully connected layers. The last layer consists of as many neurons as classes have to be separated, in our case ten. The classification is done by assigning the corresponding class of the neuron with the highest output value.

One cost function for neural networks trained with the back propagation algorithm is the mean square error (MSE) of the training set. The MSE is the mean value of the quadratic loss function $E(\alpha)$, which is given by

$$E_i(\alpha) = (d_i - f(\vec{x}_i, \alpha))^2. \quad (1)$$

In (1), α is the set of classifier parameters, d_i is the desired output for the ith element of the training set and $f(\vec{x}_i, \alpha)$ is the classifier response to input $\vec{x}_i$. The MSE of the complete training set with size N is thus

$$MSE(\alpha) = \frac{1}{N}\sum_{i=1}^{N} E_i(\alpha) = \frac{1}{N}\sum_{i=1}^{N}(d_i - f(\vec{x}_i, \alpha))^2. \quad (2)$$

The MSE is also called the empirical risk with respect to quadratic loss and classifiers using this error as a performance measure are said to implement the empirical risk minimization [5].

The training of our network is performed by the stochastic diagonal Levenberg-Marquardt algorithm that is presented in [19, 29]. The core of this algorithm is the stochastic update rule

$$\alpha_l^{(k+1)} = \alpha_l^{(k)} - \gamma_l^{(k)} \frac{\partial E_i(\alpha)}{\partial \alpha_l^{(k)}}, \quad (3)$$

where $\alpha_l^{(k)}$ is the lth element of the parameter set α at iteration k, E_i is the instantaneous loss function of (1) for image i, and $\gamma_l^{(k)}$ is the step size for the particular weight α_l at iteration k. The dependency of the step size on the iteration indicates that the step size is not fixed during the training, but is dynamically updated. The calculation of the step size is done by

$$\gamma_l^{(k)} = \frac{\eta^{(k)}}{\mu + g_l^{(k)}} \quad (4)$$

with the constant μ and a parameter $\eta^{(k)}$ that prevents the step size from becoming too large when the estimate of the second derivative $g_l^{(k)}$ of the loss function $E_i(\alpha)$ with respect to α_l is small. For the calculation of $g_l^{(k)}$ the Gauss-Newton approximation is used that guarantees a nonnegative estimate [19]. The parameter η is marked here as dependent on the iteration, but is fixed over several epochs of the training.[1]

Because of the weight sharing, the first and the second partial derivative of the loss function are sums of partial derivatives with respect to the connections that actually share the specific parameter α_l

$$\frac{\partial E_i(\alpha)}{\partial \alpha_l^{(k)}} = \sum_{(m,n)\in V_l} \frac{\partial E_i(\alpha)}{\partial w_{mn}^{(k)}}, \quad (5)$$

[1] The training of neural networks is separated into epochs; in each epoch the complete dataset is presented one time to the classifier [30].

$$\frac{\partial^2 E_i(\alpha)}{\partial \alpha_l^{2(k)}} = \sum_{(m,n)\in V_l} \sum_{(j,k)\in V_l} \frac{\partial^2 E_i(\alpha)}{\partial w_{mn}^{(k)} \partial w_{jk}^{(k)}}. \tag{6}$$

In (5) and (6) the w_{mn} are the connection weights from neuron n to m and V_l is the set of unit index pairs (m, n) such that the connection between neuron m and n share the parameter α_l, i.e.

$$w_{mn} = \alpha_l \qquad \forall\,(m, n) \in V_l. \tag{7}$$

Further details of the algorithm and the approximations that are done to compute the derivatives can be found in [19].

B. Support Vector Machines

In contrast to neural networks, SVMs follow the structural risk minimization principle, where the complexity of the classifier is additionally taken into account. The complexity of a classifier is crucial for its generalization capability and must particularly be considered if the size N of the training set is small compared to the number of features. For this reason, the Vapnik-Chervonenkis (VC)-dimension h was introduced to measure the complexity of a classifier. The lower the VC-dimension, the better the generalization of the classifier. The higher the VC-dimension, the higher the probability of overfitting to the training data [31]. To incorporate the VC-dimension into the minimization problem that has to be solved during the training, an additional term $\Phi(\frac{N}{h})$ is added to the empirical risk to define the structural risk

$$R_{struc}(\alpha, h) = R_{emp}(\alpha, h) + \Phi\left(\frac{N}{h}\right), \tag{8}$$

which is an upper bound on the generalization error. R_{emp} in (8) corresponds to the empirical risk, but this time not to the MSE of (2), which was used for neural networks, but to the specific number of misclassifications in the training set. The empirical risk is dependent on the VC-dimension because a higher VC-dimension would lead to a lower empirical risk. But in that case the second term, the confidence interval $\Phi(\frac{N}{h})$, would increase because it only depends on the ratio between the size of the training set and the VC-dimension. SVMs are designed to find the best trade-off between these two terms, decrease the empirical error while keeping the VC-dimension as low as possible. Because of this, SVMs are classifiers with a very high generalization capability.

The actual classification is done by a separating hyperplane, defined by

$$\vec{w} \cdot \vec{x} + b = 0, \tag{9}$$

where $\vec{w}$ is a learned weight vector, $\vec{x}$ is the input vector and b is the bias. For a two class problem and under the assumption that the training samples are linearly separable, correct classification is achieved if

$$\vec{w} \cdot \vec{x} + b \geq 1 \qquad for \quad y_i = +1 \tag{10}$$

$$\vec{w} \cdot \vec{x} + b \leq -1 \qquad for \quad y_i = -1. \tag{11}$$

Here y_i is the class of x_i and the distance between the input and the separating hyperplane is at least one, which can be reached by a simple rescaling of the weight vector and the bias [30]. Equations (10) and (11) can be combined to

$$y_i\left((\vec{w} \cdot \vec{x}_i) + b\right) \geq 1. \tag{12}$$

To determine the optimal separating hyperplane the function

$$\Phi(\vec{w}) = \frac{1}{2}(\vec{w} \cdot \vec{w}), \tag{13}$$

which in this context is directly related to the VC-dimension, is minimized under the constraint in (12) by quadratic optimization. The optimum condition of the minimum Euclidean norm of the weight vector provides the maximum possible separation between positive and negative examples, while the expression "$\geq$1" instead of "$>$ 0" in (12) prevents the norm of becoming arbitrarily small [33]. When the given samples are not linearly separable, nonnegative, so called slack variables ξ_i are used to define the soft-margin separating hyperplane

$$y_i\left((\vec{w} \cdot \vec{x}_i) + b\right) \geq 1 - \xi_i. \tag{14}$$

With the constraint of (14) the weight vector $\vec{w}$ is determined by the minimization of

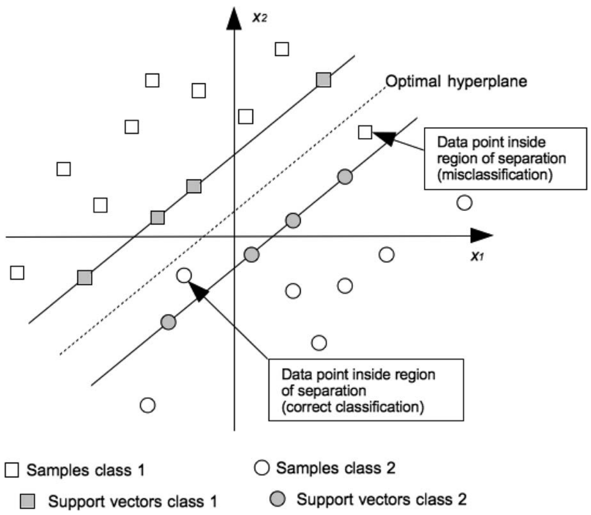
$$\Phi(\vec{w}, \xi) = \frac{1}{2}(\vec{w} \cdot \vec{w}) + C\left(\sum_{i=1}^{N} \xi_i\right), \tag{15}$$

where C is a constant [31]. In both, the separable and the nonseparable cases, the minimization problem can be solved by quadratic optimization with the use of Lagrange multipliers [30]. The first term in (15) is related to the VC-Dimension of the SVM, while the second term, known as hinge loss, is a weighting of the training errors with their distance to the separating hyperplane. The parameter C is selected experimentally by the use of a validation set during the training [30].

The basic idea of SVMs follows Cover's theorem on the separability of patterns, which states that a pattern recognition problem cast in a high-dimensional space Z nonlinearly is more likely to be linearly separable than in a low-dimensional space X [30]. To implement this principle a transformation $T(\cdot)$ of the data to a high-dimensional Hilbert space Z is performed, but this transformation is not considered in explicit form. Instead only the inner product $\langle\cdot,\cdot\rangle$ between transformed vectors must be specified. This inner product of Z can be calculated in the low-dimensional space X with a kernel $K(\vec{x}_i, \vec{x}_j)$, which follows some general conditions [31]. With this kernel an expression for the inner product of transformed vectors in Z is given by

$$\langle \vec{z}_i, \vec{z}_j \rangle = \langle T(\vec{x}_i), T(\vec{x}_j) \rangle = K(\vec{x}_i, \vec{x}_j). \tag{16}$$

Now it is possible to design a linear decision function in the high-dimensional space Z with the nonlinear



x_2
x_1
Optimal hyperplane
Data point inside region of separation (misclassification)
Data point inside region of separation (correct classification)
□ Samples class 1
○ Samples class 2
▨ Support vectors class 1
● Support vectors class 2

Fig. 3. Optimal hyperplane in a two-dimensional feature space [30].

mapping such that, for $\vec{x} \in X$,

$$f(\vec{x}) = \sum_{support\ vectors} y_i \alpha_i K(\vec{x}, \vec{x}_i) + b. \qquad (17)$$

The support vectors $\vec{x}_i$ are those training vectors, which, when transformed, lie closest to the separating hyperplane in Z and satisfy (10) and (11) with equality. The coefficients α_i are the Lagrange multipliers determined during the training.

With (17) the explicit formulation of the hyperplane in the high-dimensional space Z is not required anymore. Instead, only the distance from the unknown transformed vector $\vec{x}$ to the hyperplane is determined. The class assignment is accomplished by evaluating the sign of $f(\vec{x})$ in (17) and, therefore, on which side of the hyperplane the unknown vector is positioned in Z. In Fig. 3 the separating hyperplane is shown for a two-dimensional feature space.

Often used kernels for the transformation are polynomial kernels and Gaussian radial basis functions (RBFs). In this paper, both are used and results are shown in Section V.

A drawback of SVMs is that they can only discriminate between two classes. Since we have to discriminate between ten classes, we have to design ten SVMs, one SVM for each class, following the one vs. all principle. The assignment of the class can be done in two ways, either with a forced decision classifier or with a rejection class. For a forced decision classifier, the value of (17) is calculated for each of the ten SVMs and the class with the highest output is assigned.

For a classifier with a rejection class two criteria are used. The first one is the number of positive responses on the input image, if none or more than one SVM gives a positive result, the rejection class is assigned. In the ideal case of only one SVM giving a positive result and thus claiming membership of the image under test to the corresponding class, the output of this SVM must exceed a confidence threshold ζ [33]. With an increasing confidence threshold, the rejection rate should increase, but since images which cannot be clearly assigned are rejected as well, the error rate should decrease too. Results on both classifier structures are given in Section V.

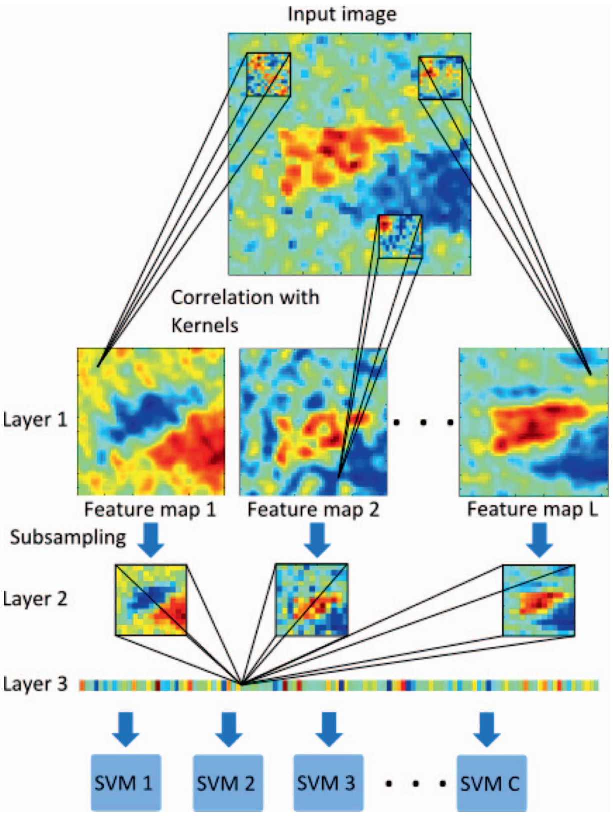


Fig. 4. Structure of the proposed ATR-System.

C. Combination of Feature Extraction and Support Vector Machines

As already mentioned in Section III.A, CNNs are separated into two parts, a feature extraction and a decision part. The network in Fig. 2 consists of a feature extraction part in the first three layers and the last two layers form the decision part. Each of these two parts can be considered as black box and can be replaced by another corresponding system. As outlined in Section III.B, SVMs possess a higher generalization capability than neural networks due to their structural advantages. Because of this, the fully connected neural network in Fig. 2 is replaced by a collection of SVMs for the final classification. Positive results for this kind of classifier were already obtained in the field of handwritten digit recognition [24, 34] and its structure is shown in Fig. 4.

With this combination, the hard part of feature selection is overcome since features are selected automatically, only the number of features has to be determined in advance or must be determined by evaluating different structures on the training data. Furthermore, the number of inputs of the SVMs is very small compared to the number of pixels, which should further improve the generalization capability according to

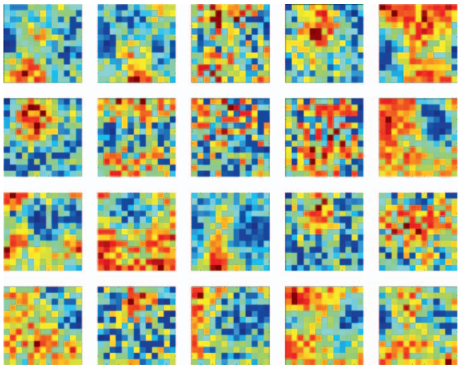


Fig. 5. Learned kernels of the first layer.

the curse of dimensionality [35]. Also, the often mentioned drawback of neural networks [36], that their behavior is hard to understand and explain, is at least partially overcome. Because the learned kernels in the first layer can be interpreted with a physical meaning, for example the feature maps of layer 1 shown in Fig. 4. The first one responds to the shadow area of the image, the second one gives the position of strong point scatterers and the last one looks like a strongly smoothed version of the image and thus gives the target area.

The structure of the CNN is fixed to twenty features in the first layer with a kernel size of 13 × 13 pixels. Since only the results of the correlation where the kernel and the image completely overlap are used, the resulting feature maps in layer 1 consist of 52 × 52 pixels for an input image of the size 64 × 64. The second layer performs a subsampling by a factor of four which results in feature maps of 13 × 13 pixels in the second layer. For the third layer, 120 kernels of the size 13 × 13 pixels are used, which leads to 120 feature maps with only one element. In this way, an input vector with 120 elements for the SVMs is generated.

Since several CNNs will be trained in the experimental part of this work, the correlation kernels of the first layer of the network that is trained only with the original data are depicted in Fig. 5. In the literature it is often reported that if the network is used for optical images these kernels are very close to Gabor filters. This seems not to be the case for SAR images, as it has also been observed in [14].

The forced decision and the classification with rejection class of the SVMs is performed in the way described in Section III.B.

IV. GENERATION OF ARTIFICIAL TRAINING DATA

To extend the training database, two techniques that generate additional images from the given data are investigated. The idea behind the first concept is to create a classifier with transformation invariance against errors and effects that probably occur in the imaging process [23]. Prominent examples for such effects are the scaling of the SAR shadow with changing depression angle, or in the case of ISAR images, the variance in the cross range scaling due to the unknown motion of the target. Because of this phenomenon, the idea is to scale the images intentionally with different factors in the range and cross range dimension. With four different scaling factors, four new images can be created out of one training image, which leads to a database five times bigger than the original one.

Another example for this kind of error is the variance in the estimation of the aspect angle of the target. The difference to the previously mentioned scaling errors is that this error occurs in the preprocessing and not in the imaging itself. But nevertheless, by presenting images that are rotated by an incorrect aspect angle during the training the invariance against this error should be increased.

A different approach is to regard SAR images as a stochastic variable from an unknown distribution. From this point of view additional data can be generated by small changes in the image and interpreted as an additional realization of a random variable. This approach is called elastic distortion [23]. Both, the affine transformation to model scaling effects in the range and cross-range and the elastic distortion are described below.

If the transformation is represented by T and the training data is organized in pairs (x_i, d_i), where x_i are the datasets and d_i are the corresponding desired responses, the generation of the new data can be described by [21]

$$(x_i, d_i) \rightarrow (T(x_i), d_i). \tag{18}$$

The basic assumption for this technique is that the desired output is invariant under the transformation.

A. Affine Transformation

For the general description of affine transformations, a destination image, denoted by $D(i, j)$, and a source image $S(p, q)$ are introduced. As source images the original images are used. To obtain a spatial size scaling in the range dimension, which is assumed to be the x-axis, the Cartesian coordinates are modified according to the relation

$$i = s_x p, \tag{19}$$

where s_x is a positive-valued scaling constant. The cross range axis stays the same. This transformation can be described in the vector-space representation with a matrix-vector multiplication in the way [37]

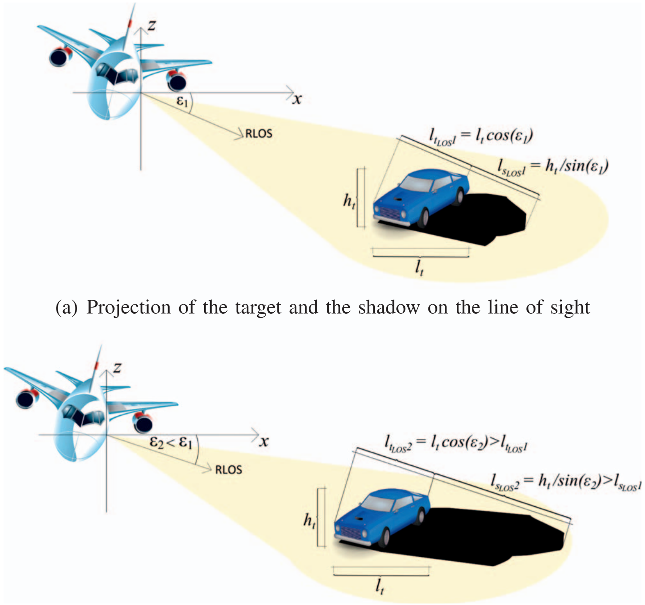
$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}. \tag{20}$$

If the cross range axis is rescaled, the transformation follows from changing the scaling parameter to the other axis

$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}. \tag{21}$$

After the rescaling an interpolation is necessary to calculate the output image, since only noninteger values from 0.9 to 1.2 are used for s_x and s_y.

In SAR images, the projection of the target area in the range direction is compressed with a factor of $\cos(\epsilon)$. The


z
x
ε_1
RLOS
$l_{tLOS1} = l_t cos(\varepsilon_1)$
$l_{sLOS1} = h_t/sin(\varepsilon_1)$
h_t
l_t
z
x
$\varepsilon_2 < \varepsilon_1$
RLOS
$l_{tLOS2} = l_t cos(\varepsilon_2) > l_{tLOS1}$
$l_{sLOS2} = h_t/sin(\varepsilon_2) > l_{sLOS1}$
h_t
l_t

(a) Projection of the target and the shadow on the line of sight

(b) Illumination with a smaller depression angle

Fig. 6. Range scaling in SAR imaging.

shadow area is scaled with a factor of $1/\sin(\epsilon)$, where ϵ is the depression angle of the SAR system [38]. This scaling for two different depression angles is shown in Fig. 6. In Fig. 6a, the projection of the target length on the line of sight of the radar depends only on the true length l_t of the target, while the projection of the shadow depends on the height h_t of the target. The given formulas follow from geometric considerations. In Fig. 6b, the same scene is illuminated with a smaller depression angle and thus both the projected target and shadow length increase.

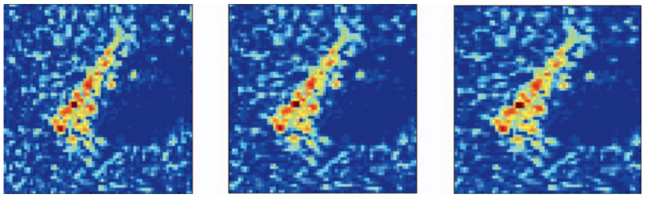
Since the used training data was measured with a depression angle of 17° and the test data with 15°, the ratio in the range scaling of the target area is given by

$$\frac{\cos(\epsilon_{test})}{\cos(\epsilon_{training})} = \frac{\cos(15^\circ)}{\cos(17^\circ)} \approx 1.01. \tag{22}$$

For the shadow area, the ratio between the test and the training images is much higher and given by

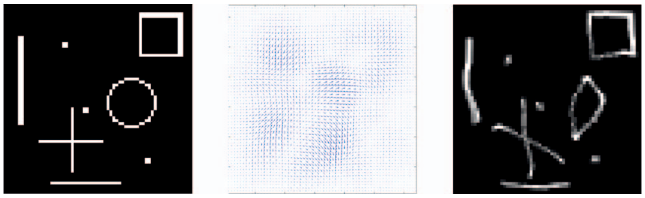
$$\frac{\sin(\epsilon_{training})}{\sin(\epsilon_{test})} = \frac{\sin(17^\circ)}{\sin(15^\circ)} \approx 1.13. \tag{23}$$

The significance of the shadow region for the classification can be seen in published classifiers like [3] or [39]. In [3], a nearest neighbor classifier is used to classify images, which consist only of the segmented shadow region of the target. In [39], the shadow profile was extracted from the image via a hidden Markov model and thus a chain code, made of the pixel-based directions in the shadow contour, represents the target shadow. Because of these results and the calculated ratios four scaling parameters are used in the range direction, from 1.05 to 1.2 with a step size of 0.05. In Fig. 7, two examples for the values $s_x = 1.1$ and $s_x = 1.2$ are shown together with the original image. The transformation was performed before the rotation described in Section II was applied. All images in Fig. 7 are in the same dynamic range. For the cross range scaling also four different values from 0.9 to 1.1 are used, although the cross range scaling is expected to be correct in both the training and the test images.



(a) Original (b) $s_x = 1.1$ (c) $s_x = 1.2$

Fig. 7. Training images with different range scaling.



(a) Original (b) Displacement (c) Distorted Image

Fig. 8. Elastic Distortion of a synthetic test scene.

To reduce the sensitivity against the estimation error of the azimuth angle, the training images are rotated on purpose by an incorrect azimuth angle. This transformation is done with a rotation matrix that is given by

$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}, \tag{24}$$

where θ is the counterclockwise angle of rotation with respect to the x-axis.

B. Elastic Distortion

To generate the image deformation, first two random displacement fields $\Delta x(x, y)$ and $\Delta y(x, y)$ are created. These random fields, which consist of random numbers between -1 and 1, generated with a uniform distribution, are then convolved with a normalized Gaussian function with standard deviation σ (in pixels). The result of this convolution is multiplied by a constant α and added to the coordinates of the original image [23]. A small standard deviation leads to a smaller smoothing effect of the random initialization of the displacement field and thus to a more distorted image. The effect of the scaling parameter α is proportional to its value. To interpolate the image to the new coordinates, an interpolation, in [40] referred to as nearest neighbor interpolation with the four closest neighbors, is used.

To visualize the deformation, an example of a test scene is shown in Fig. 8. In this test scene, several geometrical objects are visible and with the displacement field, which is visualized in the second image by a vector plot, the distorted image in Fig. 8c is produced. In this example the parameter of the elastic distortion are set to $\sigma = 5$ and $\alpha = 5$.

In Fig. 9 two elastic distortion examples are shown together with the original image, for these images the parameter of the Gaussian function are set to $\sigma = 10$ and $\alpha = 3$.

This technique was initially introduced to simulate different handwritings and to robustify classifiers for character recognition [23]. But in terms of radar images this can be interpreted as an increasing robustness against different target configurations or different articulations. This technique should also robustify the classifier against scintillation effects, since closely positioned scatterers might get merged or one small scatterer starts to widen out.

V. EXPERIMENTAL RESULTS

The classification is performed with the CNN alone and in combination with two different kernels of the SVMs, a polynomial kernel of the order three (poly3) and a RBF kernel. For all experiments the structure of the classifier is fixed to the configuration described in Section III.C. As training data the original images plus the different combinations of the additional training data are used. Four different indicators are used to compare the results of the different data sets:

- probability of correct classification: P_{CC}
- probability of error: P_{err}
- probability of rejection: P_{rej}
- P_{CC} under the condition that the images were classified at all, i.e. have not been rejected: $P_{CC|C}$

The last two probabilities are only used when a rejection class is installed. The first two probabilities and, when calculated, the third are always calculated with respect to the entire data set under test.

In the upcoming tables, the extension of the database with the affine transformation in range is marked with the letter “r”, the extension in cross-range with “cr”, and the extension with the elastic distortion images with “ed”. Each of these extensions has the same size, namely four times the size of the original data. Therefore, the datasets marked with one and two extension are five and nine times larger than the original dataset, respectively. When all the extensions are used, the dataset is thirteen times larger than the original one.

The forced decision results of the CNN alone are shown in Table II. For the combination with the polynomial and the RBF SVMs, the corresponding results are shown in Table III and Table IV, respectively. In all tables, the increased performance with the artificial training data is visible; in case of the polynomial SVM, a maximum P_{CC} of 99.5% is reached with all three artificial datasets. The RBF SVM reaches its maximum with the affine transformation in range and the elastic distortion. The results of the CNN alone are below the CNN-SVM combination for all datasets, which shows the benefit of the improved generalization. The confusion matrix of the maximum P_{CC} is depicted in Fig. 11.

TABLE II
Forced Decision Results of the CNN Classifier

Data	P_{CC}	P_{err}
original data	96.00%	4.00%
original data + r	97.56%	2.44%
original data + cr	97.63%	2.37%
original data + ed	98.06%	1.94%
original data + r + cr	97.97%	2.03%
original data + ed + r	98.0%	2.0%
original data + ed + cr	98.03%	1.97%
original data + ed + r + cr	99.13%	0.87%

TABLE III
Forced Decision Results of the CNN-SVM poly3 Classifier

Data	P_{CC}	P_{err}
original data	98.19%	1.81%
original data + r	98.63%	1.37%
original data + cr	98.66%	1.34%
original data + ed	98.91%	1.09%
original data + r + cr	98.78%	1.22%
original data + ed + r	99.09%	0.91%
original data + ed + cr	99.16%	0.84%
original data + ed + r + cr	99.5%	0.5%

TABLE IV
Forced Decision Results of the CNN-SVM RBF Classifier

Data	P_{CC}	P_{err}
original data	98.28%	1.72%
original data + r	98.72%	1.28%
original data + cr	98.66%	1.34%
original data + ed	99.06%	0.94%
original data + r + cr	98.88%	1.12%
original data + ed + r	99.31%	0.69%
original data + ed + cr	99.09%	0.91%
original data + ed + r + cr	99.25%	0.75%

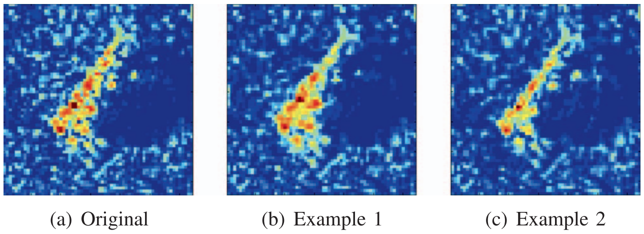

(a) Original (b) Example 1 (c) Example 2

Fig. 9. Two realizations of the elastic distortion training images.

For comparison, some results on the ten class MSTAR data of published classifiers are summarized in Table V. In [3], the images were classified by a nearest neighbor approach, but for this result the complete images were used, not only the shadow region as mentioned in Section IV.A. The classification in [41] was done with a sparse representation method. This method is based on the assumption that an unknown image can be sparsely represented by a linear combination of the known database. After the reconstruction the class with the

TABLE V
Published Forced Decision Results on Ten Class MSTAR Data

Classifier	P_{CC}	P_{err}
Nearest Neighbor [3]	86.78%	13.22%
Sparse Representation [41]	94.69%	5.31%
Gaussian Model [43]	97.18%	2.82%
Polar Mapping [43]	98.81%	1.19%

TABLE VI
Results With Rejection Class of the CNN-SVM poly3 Classifier

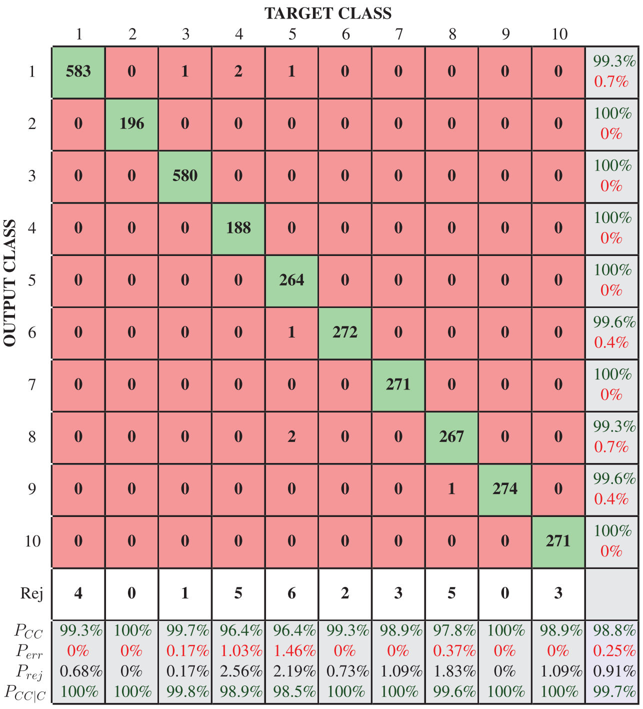
Data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
original data	93.82%	0.53%	5.65	99.44
original data + r	96.66%	0.5%	2.84%	99.49%
original data + cr	96.38%	0.56%	3.06%	99.42%
original data + ed	97.81%	0.37%	1.82%	99.62%
original data + r + cr	97.53%	0.53%	1.94%	99.46%
original data + ed + r	97.94%	0.34%	1.72%	99.65%
original data + ed + cr	97.97%	0.37%	1.66%	99.62%
original data + ed + r + cr	98.84%	0.25%	0.91%	99.75%

TABLE VII
Results With Rejection Class of the CNN-SVM RBF Classifier

Data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
original data	96.07%	0.75%	3.18%	99.23%
original data + r	97.66%	0.37%	1.97%	99.62%
original data + cr	97.56%	0.5%	1.94%	99.49%
original data + ed	98.5%	0.44%	1.06%	99.56%
original data + r + cr	97.94%	0.53%	1.53%	99.46%
original data + ed + r	98.38%	0.22%	1.4%	99.78%
original data + ed + cr	98.44%	0.41%	1.15%	99.59%
original data + ed + r + cr	98.97%	0.28%	0.75%	99.72%

smallest residual is assigned. The sparse representation is calculated for three consecutive images at the same time to use the mutual correlation between the images. In [42], a parametric signal model was used to model SAR images as Gaussian random vectors and the classification was done in a Bayesian approach by selecting the target class which maximizes the estimated posterior probability. The polar mapping algorithm in [43] performs a coordinate transformation from the Cartesian system to a polar system, where the center of the polar system is placed on the center of the target. After this transformation a profile corresponding to the shape of the target is created in the $r - \Theta -$ plane, where r and Θ are the polar coordinates. These profiles are used for classification with a nearest neighbor approach.

In Table VI and Table VII, the results with rejection class are shown for the CNN SVM combinations with a rejection threshold ζ of zero. Therefore, only the first of the two criteria described in Section III.B is used; images are rejected if none or more than one SVM gives a positive result. These results show the same trend as the forced decision classification that with an increasing amount of training data the results are getting better. The polynomial SVM has the maximum P_{CC} and also the maximum $P_{CC|C}$ with the largest amount of training data. In case of the RBF SVM, the maximum P_{CC} is achieved with all three extensions of the training data, but the conditional correct classification rate is maximized with the use of the elastic distortion and the range scaling, where also the error rate is minimized. In Fig. 10 the confusion matrix of the CNN-polynomial SVM combination with the maximum amount of training data is depicted. All performance indicators of this classification lie between the indicators of the best P_{CC} and the best $P_{CC|C}$ results of the CNN-RBF SVM combination.

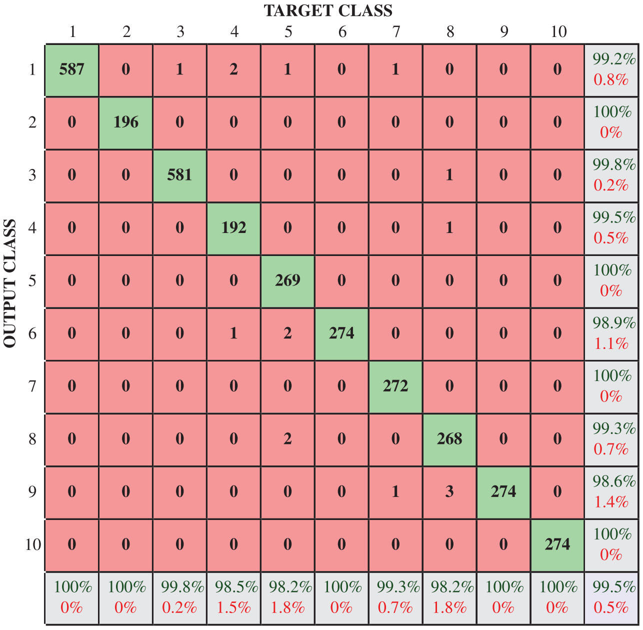


TARGET CLASS

OUTPUT CLASS

	1	2	3	4	5	6	7	8	9	10	
1	583	0	1	2	1	0	0	0	0	0	99.3% 0.7%
2	0	196	0	0	0	0	0	0	0	0	100% 0%
3	0	0	580	0	0	0	0	0	0	0	100% 0%
4	0	0	0	188	0	0	0	0	0	0	100% 0%
5	0	0	0	0	264	0	0	0	0	0	100% 0%
6	0	0	0	0	1	272	0	0	0	0	99.6% 0.4%
7	0	0	0	0	0	0	271	0	0	0	100% 0%
8	0	0	0	0	2	0	0	267	0	0	99.3% 0.7%
9	0	0	0	0	0	0	0	1	274	0	99.6% 0.4%
10	0	0	0	0	0	0	0	0	0	271	100% 0%
Rej	4	0	1	5	6	2	3	5	0	3	
P_{CC}	99.3%	100%	99.7%	96.4%	96.4%	99.3%	98.9%	97.8%	100%	98.9%	98.8%
P_{err}	0%	0%	0.17%	1.03%	1.46%	0%	0%	0.37%	0%	0%	0.25%
P_{rej}	0.68%	0%	0.17%	2.56%	2.19%	0.73%	1.09%	1.83%	0%	1.09%	0.91%
$P_{CC\|C}$	100%	100%	99.8%	98.9%	98.5%	100%	100%	99.6%	100%	100%	99.7%

Fig. 10. Confusion matrix CNN-SVM poly3 with maximum training data.



TARGET CLASS

OUTPUT CLASS

	1	2	3	4	5	6	7	8	9	10	
1	587	0	1	2	1	0	1	0	0	0	99.2% 0.8%
2	0	196	0	0	0	0	0	0	0	0	100% 0%
3	0	0	581	0	0	0	0	1	0	0	99.8% 0.2%
4	0	0	0	192	0	0	0	1	0	0	99.5% 0.5%
5	0	0	0	0	269	0	0	0	0	0	100% 0%
6	0	0	0	1	2	274	0	0	0	0	98.9% 1.1%
7	0	0	0	0	0	0	272	0	0	0	100% 0%
8	0	0	0	0	2	0	0	268	0	0	99.3% 0.7%
9	0	0	0	0	0	0	1	3	274	0	98.6% 1.4%
10	0	0	0	0	0	0	0	0	0	274	100% 0%
	100% 0%	100% 0%	99.8% 0.2%	98.5% 1.5%	98.2% 1.8%	100% 0%	99.3% 0.7%	98.2% 1.8%	100% 0%	100% 0%	99.5% 0.5%

Fig. 11. Confusion matrix best forced decision result.

TABLE VIII
Results With Rejection Class of the CNN-SVM RBF Classifier

ζ	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
0	98.97%	0.28%	0.75%	99.72%
0.2	98.5%	0.09%	1.4%	99.91%
0.4	97.78%	0.06%	2.15%	99.94%
0.6	96.35%	0.06%	3.59%	99.94%
0.8	93.88%	0.03%	6.09%	99.97%
1.0	89.7%	0.03%	10.27%	99.97%
1.06	87.7%	0.0%	12.3%	100%
1.2	81.7%	0.0%	18.3%	100%

TABLE IX
Results CNN-SVM RBF Classifier for $\phi = 2°$ With Rejection

Training data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
0°	95.22%	0.91%	3.87%	99.06%
± 3°	96.35%	0.69%	2.97%	99.29%
± 6°	98.16%	0.28%	1.56%	99.71%
± 9°	97.88%	0.47%	1.65%	99.52%
± 12°	98.0%	0.22%	1.78%	99.78%
± 15°	97.72%	0.59%	1.69%	99.4%

For the classification with a variable rejection threshold ζ the CNN-SVM RBF combination with the maximum amount of training data is used. This combination gave the best absolute correct classification rate in Table VII. The results are shown in Table VIII and already for $\zeta = 0.2$ the error rate has dropped to three out of 3203 images, while P_{CC} is still at 98.5%. For an error rate of 0%, the rejection threshold must be increased to 1.06, which leads to a rejection rate of a little bit more than 12%.

For the last experiment, the given azimuth angles of the test data are corrupted by Gaussian noise with different standard deviations ϕ. With this modification, a more realistic scenario should be created since the estimation of the aspect angle of targets is an ongoing challenge. The azimuth angles of the training data are kept fixed, but additionally every image is copied several times and rotated by a different angle. For every new training set new images with a change of ± 3° compared to the last one is added to the existing set. This means the first training set consists only of the training data, for the second network the original images and images rotated by the given azimuth angle ± 3° are used as training data. The training data of the third network is made up of the original data, the original data rotated by ± 3° and by ± 6°. This procedure continues until the maximum rotation of ± 15° is reached, which results in six different networks. The rejection threshold is again set to zero and the structure is the same as described in Section III.C.

Results of the CNN-SVM RBF classifier, trained with the different datasets, for azimuth standard deviations of 2, 4, 6, and 8 are shown in Tables IX–XII.

TABLE X
Results CNN-SVM RBF Classifier for $\phi = 4°$ With Rejection

Training data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
0°	92.60%	1.06%	6.34%	98.87%
± 3°	95.25%	0.94%	3.81%	99.03%
± 6°	97.66%	0.41%	1.94%	99.59%
± 9°	97.91%	0.44%	1.65%	99.56%
± 12°	97.97%	0.31%	1.72%	99.68%
± 15°	97.5%	0.5%	2.0%	99.49%

TABLE XI
Results CNN-SVM RBF Classifier for $\phi = 6°$ With Rejection

Training data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
0°	88.42%	1.44%	10.15%	98.4%
± 3°	92.44%	1.25%	6.31%	98.67%
± 6°	96.82%	0.47%	2.72%	99.52%
± 9°	97.69%	0.34%	1.97%	99.65%
± 12°	97.81%	0.25%	1.94%	99.75%
± 15°	97.75%	0.44%	1.81%	99.55%

TABLE XII
Results CNN-SVM RBF Classifier for $\phi = 8°$ With Rejection

Training data	P_{CC}	P_{err}	P_{rej}	$P_{CC\|C}$
0°	81.24%	3.84%	14.92%	95.49%
± 3°	86.26%	2.47%	11.27%	97.22%
± 6°	93.72%	1.0%	5.28%	98.95%
± 9°	96.5%	0.72%	2.78%	99.26%
± 12°	97.25%	0.44%	2.31%	99.55%
± 15°	97.22%	0.47%	2.31%	99.52%

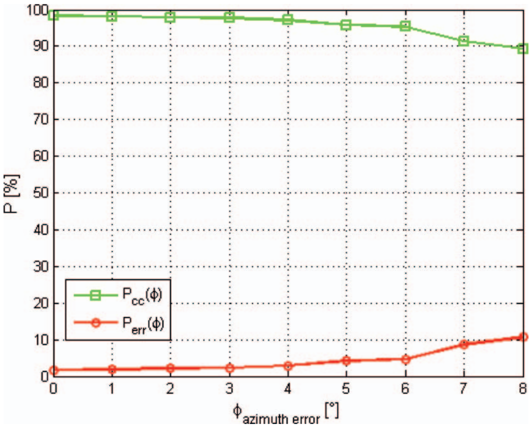


Fig. 12. Result CNN-SVM RBF classifier with azimuth error and original training data.

These results show the increased performance with the rotated images used as a database. Especially the network trained with the images rotated up to ± 12° shows a very good performance over all test datasets. The results of the forced decision classification are depicted in Fig. 12 and Fig. 13. In Fig. 12 the performance of the CNN-SVM RBF combination trained only with the original data is shown and Fig. 13 shows the results of the classifier trained with the up to ± 12° rotated images. Comparison of the two figures shows the stable performance of the classifier trained with the increased database over all values of azimuth error, while the classifier with the original training data drops down for high errors.

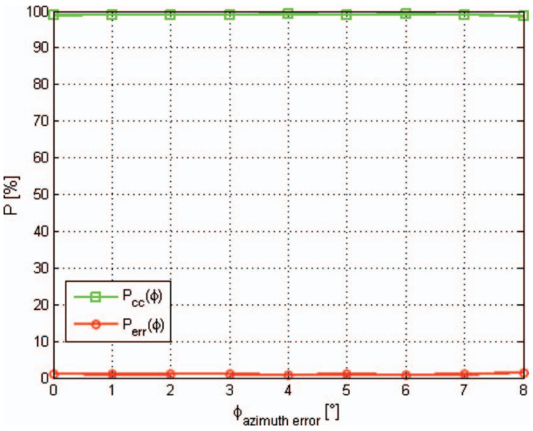


Fig. 13. Result CNN-SVM RBF classifier with azimuth error and extended training data.

VI. CONCLUSION

The presented combination of a convolutional neural network and support vector machines was evaluated using the ten class MSTAR database. With the presented training methods a correct classification rate of 99.5% in the forced decision case is achieved. With rejection class, the error rate is decreased below 0.3% while the correct classification rate is kept close to 99%. The transformations that should represent typical imaging errors increase the performance, but it seems that the highest benefit comes from the elastic distortion that produces new data by small random deformations of known images. With the presented confidence measure an error rate of zero is achieved with a correct classification rate of 87.7% for a classification with rejection class.

The results with incorrect azimuth angle of the targets clearly show the robust behavior of the classifier after the training with the proposed training method and overcomes its rotational variance up to a certain degree.

ACKNOWLEDGMENTS

The author would like to thank the anonymous reviewers and the associate editor Alex Charlish for their helpful comments. Special thanks go to Franz Dommermuth and Tanja Bieker for the careful reading of the manuscript and helpful feedbacks.

REFERENCES

[1] Rosenbach, K., and Schiller, J.
Identification of aircraft on the basis of 2-D radar images.
In *Proceedings of the Record of the IEEE 1995 International Radar Conference*, May 1995, 405–409.

[2] Haumtratz, T., Worms, J., and Schiller, J.
Classification of air targets including a rejection stage for unknown targets.
In *Proceedings of the 2010 11th International Radar Symposium (IRS)*, June 2010, 1–4.

[3] Schumacher, R., and Schiller, J.
Non-cooperative target identification of battlefield targets—Classification results based on SAR images.
In *Proceedings of the 2005 IEEE International Radar Conference*, May 2005, 167–172.

[4] Chadwick, J., and Williams, G. L.
Air target identification—Concept to reality.
In *Proceedings of the 2007 IET International Conference on Radar Systems*, Oct. 2007, 1–5.

[5] Zhao, Q., Principe, J. C., Brennan, V. L., Xu, D., and Wang, Z.
Synthetic aperture radar automatic target recognition with three strategies of learning and representation.
Optical Engineering, **39**, 5 (2000), 1,230–1,244.

[6] Bieker, T.
Statistical evaluation of decision-level fusion methods for non-cooperative target identification by radar signatures.
In *Proceedings of the 2008 11th International Conference on Information Fusion*, June 2008, 1–7.

[7] Jacobs, S. P., and O'Sullivan, J. A.
Automatic target recognition using sequences of high resolution radar range-profiles.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 364–381.

[8] de Wit, J. J. M., Harmanny, R. I. A., and Molchanov, P.
Radar micro-Doppler feature extraction using the singular value decomposition.
In *Proceedings of the 2014 International Radar Conference (Radar)*, Oct. 2014, 1–6.

[9] Molchanov, P., Egiazarian, K., Astola, J., Totsky, A., Leshchenko, S., and Jarabo-Amores, M. P.
Classification of aircraft using micro-Doppler bicoherence-based features.
IEEE Transactions on Aerospace and Electronic Systems, **50**, 2 (April 2014), 1455–1467.

[10] Tait, P.
Automatic recognition of air targets.
In *Radar Automatic Target Recognition (ATR) and Non-Cooperative Target Recognition (NCTR)*, number 33 in Radar, Sonar and Navigation Series, ch. 3, David Blacknell and Hugh Griffiths, Eds. Stevenage, UK: The Institution of Engineering and Technology, 2013, pp. 37–75.

[11] Schachter, B. J.
Target classification strategies.
In *Proceedings of SPIE*, Vol. 9476, 2015, pp. 947602.

[12] Morgan, D. A. E.
Deep convolutional neural networks for ATR from SAR imagery.
In *Proceedings of SPIE*, Vol. **9475**, 2015, pp. 94750F.

[13] Chen, S., and Wang, H.
SAR target recognition based on deep learning.
In *Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2014, 541–547.

[14] Profeta, A., Rodriguez, A., and Clouse, H. S.
Convolutional neural networks for synthetic aperture radar classification.
In *Proceedings of SPIE*, Vol. 9843, 2016, pp. 98430M.

[15] Wilmanski, M., Kreucher, C., and Lauer, J.
Modern approaches in deep learning for SAR ATR.
In *Proceedings of SPIE*, Vol. 9843, 2016, pp. 98430N.

[16] Ødegaard, N., Knapskog, A. O., Cochin, C., and Louvigne, J.-C.
Classification of ships using real and simulated data in a convolutional neural network.
In *Proceedings of the 2016 IEEE Radar Conference (RadarConf.)*, May 2016.

[17] Wagner, S.
Combination of convolutional feature extraction and support vector machines for radar ATR.
In *Proceedings of the 2014 17th International Conference on Information Fusion (FUSION)*, July 2014.

[18] Wagner, S.
Increasing the performance of ATR systems using artificial training data.
In *Proceedings of the NATO Specialists' Meeting SET-228 on*

Radar Imaging for Target Identification, Pisa, Italy, Oct. 2015, 7-1–7-12.

[19] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**, 11 (Nov. 1998), 2,278–2,324.

[20] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 8 (Aug. 2013), 1,798–1,828.

[21] Niyogi, P., Girosi, F., and Poggio, T. Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, **86**, 11 (Nov 1998), 2,196–2,209.

[22] Zhao, Q., and Principe, J. C. Improving ATR performance by incorporating virtual negative examples. In *Proceedings of the International Joint Conference on Neural Networks IJCNN*, 1999, **5**, 3,198–3,203.

[23] Simard, P. Y., Steinkraus, D., and Platt, J. C. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Aug. 2003, 958–963.

[24] Lauer, F., Suen, C. Y., and Bloch, G. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, **40**, 6 (2007), 1,816–1,824.

[25] Gil-Pita, R., Jarabo-Amores, P., Rosa-Zurera, M., and Lopez-Ferreras, F. Improving neural classifiers for ATR using a kernel method for generating synthetic training sets. In *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*, 2002, 425–434.

[26] Diemunsch, J. R. and Wissinger, J. Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: Search technology for a robust ATR. In *Proceedings of SPIE*, **3370**, 1998, 481–492.

[27] Wehner, D. R. *High-Resolution Radar* (2nd ed.). Artech House, 1995.

[28] Meth, R., Chellappa, R., and Kuttikkad, S. Target aspect estimation from single and multi-pass SAR images. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1998, **5**, 2,745–2,748.

[29] le Cun, Y. Generalization and network design strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, June 1989.

[30] Haykin, S. *Neural Networks and Learning Machines* (3rd ed.). Upper Saddle River, NJ: Pearson Education Inc., 2009.

[31] Vapnik, V. N. Statistics for Engineering and Information Science. *The Nature of Statistical Learning Theory* (2nd ed.). New York: Springer-Verlag, 2000.

[32] Bengio, Y. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, **2**, 1 (2009), 1–127.

[33] Schölkopf, B. and Smola, A. J. *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[34] Niu, X.-X., and Suen, C. Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, **4** (2012), 1,318–1,325.

[35] Jain, A. K., Duin, R. P. W., and Mao, J. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 1 (Jan. 2000), 4–37.

[36] Schiller, J., Capabilities developments and challenges in non-cooperative target identification using radar. In *Proceedings of the Microwaves, Radar and Remote Sensing Symposium (MRRS)*, Aug. 2011, 24–27.

[37] Pratt, W. K. *Digital Image Processing* (4th ed.). John Wiley & Sons, 2007.

[38] Keydel, E. R., Lee, S. W., and Moore, J. T. MSTAR extended operating conditions: a tutorial. *Proceedings of SPIE*, **2757**, 1996, 228–242.

[39] Papson, S., and Narayanan, R. M. Classification via the shadow region in SAR imagery. *IEEE Transactions on Aerospace and Electronic Systems*, **48**, 2 (Apr. 2012), 969–980.

[40] Özdemir, C. *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*. John Wiley & Sons, 2012.

[41] Zhang, H., Nasrabadi, N. M., Zhang, Y., and Huang, T. S. Multi-view automatic target recognition using joint sparse representation. *IEEE Transactions on Aerospace and Electronic Systems*, **48**, 3 (July 2012), 2,481–2,497.

[42] O'Sullivan, J. A., DeVore, M. D., Kedia, V., and Miller, M. I. SAR ATR performance using a conditionally gaussian model. *IEEE Transactions on Aerospace and Electronic Systems*, **37**, 1 (Jan. 2001), 91–108.

[43] Park, J.-I., and Kim, K.-T. Modified polar mapping classifier for SAR automatic target recognition. *IEEE Transactions on Aerospace and Electronic Systems*, **50**, 2 (Apr. 2014), 1,092–1,107.

Simon Wagner received the B.Eng. and the M.Sc. degree in electrical engineering from the University of Applied Sciences in Trier, Germany, in 2010 and 2013, respectively.

He joined the Fraunhofer FHR in 2012 for his Master Thesis and since 2013 he has been a research scientist in the NCI group of the cognitive radar department. His research interests include deep learning and sparse representation for target classification purposes. He is currently pursuing the Ph.D. degree at the University of Siegen in Germany.