SAMPLE Dataset Objects Classification Using Deep Learning Algorithms

Michal TURCANIK [1], *Jozef PERDOCH* [2]

[1] Dept. of Informatics, Armed Forces Academy of gen. M. R. Štefánik, Demänová 393, 03101 Liptovský Mikuláš, Slovakia
[2] Dept. of Electronics, Armed Forces Academy of gen. M. R. Štefánik, Demänová 393, 03101 Liptovský Mikuláš, Slovakia

michal.turcanik@aos.sk, jozef.perdoch@aos.sk

Submitted October 6, 2022 / Accepted December 15, 2022 / Online first January 27, 2023

Abstract. *The main topic of the article is automatic target classification of the synthetic aperture radar images based on the dataset composed of measured and synthetic data. The original contribution of the authors is their own topology of the convolutional neural network (CNN) with 1, 2, 3, and 4 tiers. The original convolutional neural network is used to classify radar images from the Synthetic And Measured Paired and Labeled Experiment (SAMPLE) dataset which consists of SAR imagery from publicly available datasets and well-matched synthetic data. The presented topologies of the CNN with 1, 2, 3, and 4 tiers were analyzed in 3 different scenarios: trained on the basis of real measured data and tested by synthetic data, trained on the basis of synthetic data, and tested by real measured data, and in the last case training and testing sets were formed by combining real measured and synthetic data. Based on the results of testing we could not use the proposed convolutional neural network trained with real measured data to classify synthetic radar images and vice versa (the 1st and the 2nd scenarios). The only last scenario with a combination of real measured and synthetic data in the training, validation, and testing data sets generates excellent results. The authors also present some confusion matrixes, which can explain the reasons for the misclassification of radar images of military equipment. Comparing achieved results with another SAMPLE dataset classification results we can prove the usability of proposed and tested CNN structures for automatic target classification of the synthetic aperture radar images. The classification accuracy of the original convolutional network is 96.1%, which is better than the results of the other research so far.*

Keywords

Synthetic aperture radar, synthetic data, SAMPLE dataset, convolutional neural networks

1. Introduction

Synthetic aperture radar (SAR) sensing is a unique technique that allows users to remotely map the reflectivity of environments or objects in high resolution through transmitting and receiving electromagnetic (EM) signals [1], [2], [3]. Radar's most important feature for SAR applications is that its relatively long wavelengths penetrate clouds and dust, and it can sense independently of most weather conditions [4], [5].

The publicly available Moving and Stationary Target Acquisition and Recognition (MSTAR) synthetic aperture radar dataset [6] has been a valuable tool in the evolution of SAR automatic target recognition (ATR) algorithms over the past two decades [7]. Because of the large number of target configurations, possible radar parameters, and environmental conditions, the SAR operating condition (OC) scale is very large. This leads to the impossible task of collecting sufficient measured data to cover the entire OC space. Thus, synthetic data must be created to augment datasets. The study of synthetic data fidelity for classification tasks is not an easy mission. To that end, the Synthetic And Measured Paired and Labeled Experiment (SAMPLE) dataset was created and introduced [7], which consists of SAR imagery from the MSTAR dataset and well-matched synthetic data. By matching sensor parameters and target configurations among the measured and synthetic data, the SAMPLE dataset is ideal for investigating the differences between measured and synthetic SAR imagery [7], [8], [9].

The convolutional neural networks (CNN) are the results of the study of the brain's visual cortex, and they have been used in image recognition since the 1980s. In the last decade, thanks to the increase in computational power, and the big variety of training data, convolutional neural networks achieve very high performance on some complex visual tasks. They are used in image search services [10], [11], self-driving cars [12], [13], automatic video classifi-

DOI: 10.13164/re.2023.0063

cation systems, medical applications [14], [15], malware classification [16], filters as bitmap objects classification [17], and others. Convolutional neural networks are not restricted to the perception of visual information, and they are strong at other tasks. The efficiency and high success rate of correct recognition of the CNN is the main reason why we used them for SAR image classification. CNN is able to recognize correctly even images that are dilated and rotated, possibly having a different size than the training samples.

Some experiments for SAMPLE dataset object classification using CNN were described in [7] and [8]. Authors in [7] realized classification experiments using a convolutional neural network (CNN) with four convolutional layers and four fully connected layers. Their experiments were mainly focused on classifying measured data from synthetic training data and comparing achieved results with different mixtures of synthetic and measured data in the training dataset. The first experiment in [7] is a useful investigation of how to effectively augment limited measured training data with synthetic data. Such investigations are of interest because, relative to the entire OC space of SAR, little measured data is available for training ATR algorithms. By systematically excluding measured data from the training set and replacing it with synthetic data in this way, much can be learned that can carry over to using synthetic data to augment the OCs necessary to enhance classification performance [7]. The second experiment in [7] is a practical investigation of how classification networks perform when no measured data is available for specific classes. This is an interesting problem because it requires innovation to successfully perform cross-domain transfer learning in a way that encourages the algorithm to know something about SAR imagery in general [7]. Authors in [7] just closely reached the classification accuracy of 95% for $k = 0.5$, i.e., the training set for each class contains 50% measured data and 50% synthetic data.

All experiments in [8] were performed using the DenseNet architecture [18] implemented in PyTorch. DenseNet is a densely connected CNN that extends the ideas of the ResNet network [19]. Authors compared the accuracy of SAMPLE dataset object classification when using the speckled, despeckled, quantized, and clutter transfer datasets [8]. Authors in [8] for $k = 0.5$ achieved the best results of classification accuracy (almost 95%) when they used speckled data and employed the clutter transfer technique.

The main goal of our experiments was to prove or deny the possibilities of various combinations for CNN training and utilize our own designed CNN structures for SAMPLE dataset object classification. The next goal is to find the best topology for our own designed CNN structures.

The paper is organized as follows. MSTAR and SAMPLE datasets are described in Sec. 2. Our own designed and used CNN structures are explained in Sec. 3. Experiments and partial results are described in Sec. 4. The paper concludes with a summary in Sec. 5.

2. MSTAR and SAMPLE Dataset

The MSTAR dataset [6] is one of the most comprehensive measured SAR datasets available to the research community. It consists of a range of one-foot resolution SAR images collected by the U. S. Air Force Research Laboratory, Sandia National Laboratory, and The Defense Advanced Research Projects Agency (DARPA) during the second half of the 1990s. The MSTAR dataset has been useful for researchers who are interested in automatic target recognition (ATR) tasks [7]. Essential parameters, which are consistent across the MSTAR dataset, are shown in Tab. 1 [6], [7].

SAMPLE dataset was constructed of measured SAR images from the MSTAR dataset [6] and simulated SAR images based on computer-aided design (CAD) models [7]. Prediction of the electromagnetic signatures of a target was realized by asymptotic ray-tracing techniques [7]. The simulated models are based on metadata that was recorded during the MSTAR Program, enabling authors to position the CAD models in the same way that the measured vehicles were placed during the collection. Thus, the authors in [7] were sufficiently able to match target and radar operat-

Parameter	Value
Range resolution	0.30 m
Range pixel spacing	0.20 m
Range extent	25.8 m
Cross-range resolution	0.30 m
Cross-range pixel spacing	0.20 m
Cross-range extent	25.8 m
Bandwidth	591 MHz
Center frequency	9.6 GHz
Image size	128 × 128
Polarization	HH
Elevations	from 14° to 18°
Taylor weighting	–35 dB

Tab. 1. Values of parameters from the MSTAR dataset [6], [7].

Vehicle (object)	Serial number
2S1	B01
BMP2	9563
BTR70	C71
M1	0AP00N
M2	MV02GX
M35	T839
M548	C245HAB
M60	3336
T72	812
ZSU23-4	D08

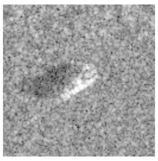
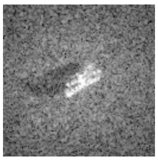
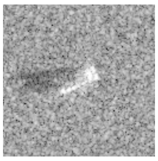
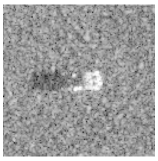
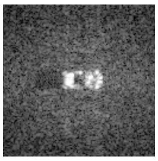
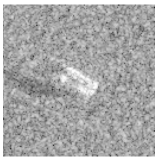
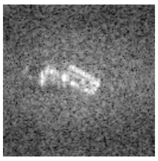
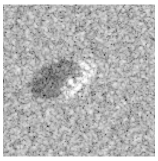
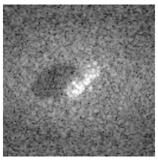
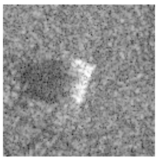
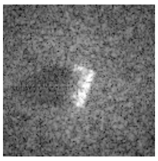
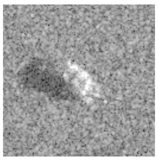
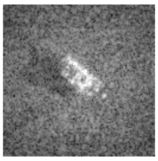
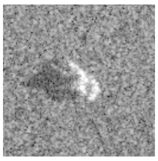
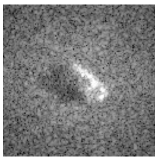
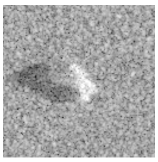
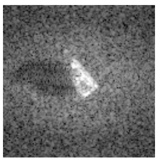
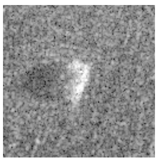
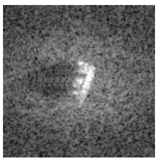
Tab. 2. List of the vehicles and their serial numbers included in the SAMPLE dataset.

ing conditions. SAMPLE dataset consists of matched measured and computer-generated (synthetic) SAR data as complex image data (in MATLAB .mat format), and magnitude-only images (in .png format) for the ten targets referred in Tab. 2.

SAMPLE dataset was approved for public release (case number: 88ABW-2019-1300, reference number: APRS-RY-19-1024) to provide a common dataset to facilitate collaboration across research organizations. For each measured SAR image, the corresponding matched synthetic image was generated. The number of samples for each target class within the publicly released SAMPLE dataset is shown in Tab. 3. SAMPLE dataset SAR .png examples (128 × 128 pixels) of matched measured and synthetic data are shown in Tab. 4.

Target class	Measured data	Synthetic data	Total
2S1	879	879	1758
BMP2	502	502	1004
BTR70	504	504	1008
M1	729	729	1458
M2	724	724	1448
M35	729	729	1458
M548	730	730	1460
M60	874	874	1748
T72	503	503	1006
ZSU23	876	876	1752
Total	**7050**	**7050**	**14100**

Tab. 3. Distribution of publicly released SAMPLE dataset for each class (vehicle).

Object class and sensing angle values	Measured data	Synthetic data	Class and sensing angle values	Measured data	Synthetic data
2S1 Elevation: 16° Azimuth: 30°			**M35** Elevation: 14° Azimuth: 20°		
BMP2 Elevation: 16° Azimuth: 180°			**M548** Elevation: 14° Azimuth: 330°		
BTR70 Elevation: 16° Azimuth: 45°			**M60** Elevation: 15° Azimuth: 70°		
M1 Elevation: 17° Azimuth: 320°			**T72** Elevation: 16° Azimuth: 140°		
M2 Elevation: 14° Azimuth: 310°			**ZSU23** Elevation: 15° Azimuth: 70°		

Tab. 4. SAMPLE dataset SAR .png examples (128 × 128 pixels) of matched measured and synthetic data [9].

SAMPLE dataset contains data for 10 targets over a 360-degree azimuth sweep and a 5-degree elevation angle sweep (14 degrees to 18 degrees). The target set contains data from the 2S1, BMP2, BTR70, M1, M2, M35, M548, M60, T72, and ZSU23 military vehicles [7], [9].

3. Proposed Convolutional Neural Networks for SAR Objects Classification

The complete architecture of the convolutional neural network for pattern recognition of the radar images is shown in Fig. 1. Proposed and used CNN consists of the following layers:

- an image input layer,
- 1st tier (consisting of a convolutional 2D layer, a batch normalization layer, a ReLU layer, and a max-pooling layer),
- 2nd tier (consisting of a convolutional 2D layer, a batch normalization layer, a ReLU layer, and a max-pooling layer),
- 3rd tier (consisting of a convolutional 2D layer, a batch normalization layer, a ReLU layer, and a max-pooling layer),
- 4th tier (consisting of a convolutional 2D layer, a batch normalization layer, a ReLU layer, and a max-pooling layer),
- a fully connected layer,
- a softmax layer,
- a classification layer.

An input image for classification enters a convolutional neural network through an image input layer. The size of an input image defines the size of the image input layer (128 × 128 pixels of grayscale picture). The input 2-dimensional (2-D) images have 256 shades of gray (8 bits) [6], [7], [9].

A convolutional layer, the most important block of a convolutional neural network, executes a convolution operation to the data received from the input layer and passes the result to the next layer. A convolution converts all the pixels in its receptive field into a vector. The most used type of convolution is the 2D convolution layer. A convolutional layer of the CNN uses a kernel (sometimes called filter) to realize convolution over the 2-dimensional input data, executing a multiplication over each element. As a result, it will be summing up the results into a single output pixel. The kernel will perform the same operation for every location it slides over, transforming a 2D matrix of features into a different 2D matrix of features.

A batch normalization layer is used to normalize data (results of the previous layer) across all observations for every input line independently. The batch normalization layers were used to speed up the training of the convolutional neural networks. This layer can also reduce the sensitivity to the initial setting of the CNN. This layer is located between convolutional layers and nonlinearities, such as rectified linear unit (ReLU) layers.

ReLU is the most used activation function in convolutional neural networks [18]. The function returns 0 for any negative input, but for any positive value x, it gives that same value back as a result. So, it can be written as:

$$f(x) = \max(0, x). \qquad (1)$$

A ReLU layer is applied after a batch normalization layer. A max-pooling layer is used for downsampling the input data in CNNs to reduce the computational load, the memory usage, and the number of parameters. This layer realizes a calculation of the maximum value for each patch of the input image matrix. A max-pooling operation highlights the most significant feature in the selected region which is better in practice than average pooling for computer vision tasks.

A fully connected layer is found in the last few layers in the network. The input to the fully connected layer is the output from the final max-pooling layer. A fully connected layer realizes the multiplication of the inputs by a weight matrix and then adds a bias vector the same way as it is realized by feed-forward neural networks.

A softmax layer realizes a softmax function to normalize the value of the input data on the base of the channel dimension. The softmax function sums all of them into one. The normalization realized by the softmax function will ensure that the sum of the components of the output vector is equal to 1. An output of the softmax function represents a probability distribution.

A classification layer defines several classes of input images. In our case, the number is 10, which corresponds to a defined number of types of military vehicles to classify. For a SAR images classifier, four topologies of convolutional neural networks were created, and their architectures are defined in Tab. 5.

Topology/ layer	Topology A	Topology B	Topology C	Topology D
Image input layer	X	X	X	X
1st tier	X	X	X	X
2nd tier		X	X	X
3rd tier			X	X
4th tier				X
Fully connected layer	X	X	X	X
Softmax layer	X	X	X	X
Classification layer	X	X	X	X

Tab. 5. Architectures of tested CNN for SAMPLE dataset objects classification.

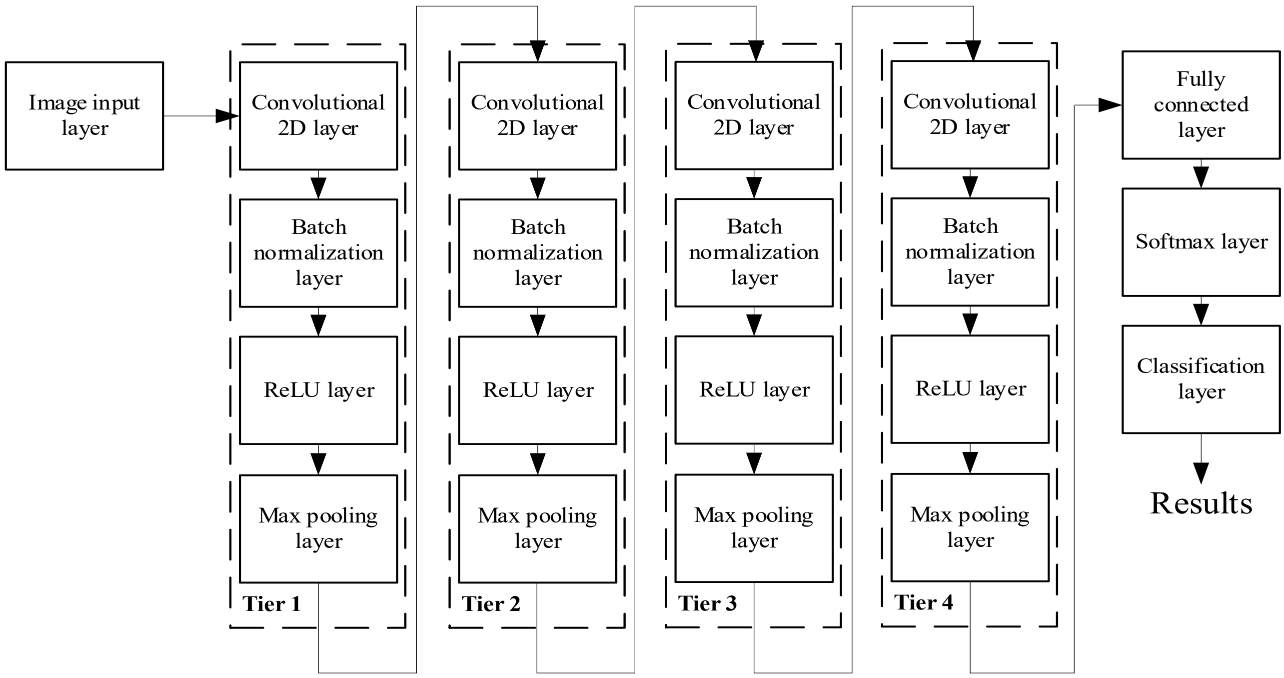


Fig. 1. The architecture of the used Convolutional Neural Network.

The difference between the individual topologies is in the number of so-called "tiers". Every tier is composed of the same number and type of layers. Topology A contains an image input layer, the 1st tier, a fully connected layer, a softmax layer, and a classification layer. Topology B consists of the same layers but in addition convolutional neural network also has the 2nd tier. Similarly, topology C contains the same layers as the previous one and the 3rd tier in addition. Topology D is composed of all four tiers and common layers (an image input layer, a fully connected layer, etc.).

The architecture of the proposed convolutional neural network was created in the Matlab development environment and Deep Learning Toolbox was used. A pre-trained network can be used for most deep-learning applications after adaptation for user data. In our case, we created and trained convolutional neural networks from scratch using the trainNetwork function. The complete architecture of the CNN was defined by layers parameter and training options were defined by the options parameter of the given function.

Our goal is to find the best topology for CNN. Generally, there is no rule to set exactly the number of layers and the number of neurons for a specific task. We will create and train four CNNs with a different number of tiers and we will analyze their efficiency. All of them will be trained with the same set of input images. The proposed topology of CNN will be evaluated with an identical set of testing images for all CNN. After that, we will choose the best CNN topology for SAR objects classification with the optimal number of tiers and neurons.

4. Analysis of Using CNNs for SAMPLE Dataset Objects Classification

Measured data from the real world are difficult to obtain and they are expensive. Sometimes it is not possible to obtain such data due to the nature and type of technology, which is under a certain degree of secrecy. Today, in many cases measured data are supplemented with synthetic data. This way we can rapidly increase the amount of existing data to create more meaningful observations of selected objects. Synthetic data generation compared to measured data is faster, more flexible, and more scalable. Using synthetic data can also be an effective manner to model and generate data that could not be measured for a different reason.

The analysis of the use of CNN, the topology, and properties that have been described above, for objects classification of SAR images, is based on the verification of three hypotheses:

1) H1: The generated CNN can be trained with measured data and the verification of the learning process results can be performed based on testing using synthetic data,
2) H2: The generated CNN can be trained with synthetic data and the verification of the learning process results can be performed by testing with the measured data,
3) H3: The generated CNN can be trained based on a set of images, which is formed by combining real meas-

ured and synthetic data. Verification of the learning process results can be performed by testing the same set (real measured and synthetic data).

For all hypotheses, the intersection between the training and test set is 0. Images that have been used in the learning process will not be used in the testing process and vice versa.

4.1 CNN Trained by Measured Data

To verify the first hypothesis, four CNN topologies with one to four tiers (a complete structure defined in Tab. 5) were developed. Measured data (7050 images) were used for training. The input set was split at a usually used ratio of 70:30 [20], where 70 percent of the samples were used to form the training set. The remaining 30 percent was used to create a validation set that was used to verify the quality of CNN's learning [20]. A complete set of synthetic data containing 7050 images (Tab. 4) was used for testing. The results obtained by a given CNN topology with a different number of tiers are shown in Fig. 2.

Classification accuracy, during training by measured data, was improved by increasing the number of tiers. CNN with one tier reached a classification accuracy of 12%, CNN with two tiers 82.4%, and CNN with three and four tiers reached a classification accuracy of more than 96%. These results were promising but testing CNN with synthetic data turned out very bad. Classification accuracy of CNN with one tier reached 8.4%, with two tiers 30%, with three tiers 34.7%, and with four tiers reached 27%. Results for the validation dataset reached using the CNN with 3 tiers are shown in Fig. 3, where we can find individual achievements for all object classes. For better results understanding, a confusion matrix was used. The meaning of particular items of rows of the confusion matrix represents the predicted class (Output Class) and the columns represent the true class (Target Class). The cells in the diagonal represent the number of objects that are correctly classified. The off-diagonal cells represent incorrectly classified objects from the given set.

A 2S1 object is presented in the first row, where the value in the first column represents the number of correctly classified images. It means that object 2S1 was classified 255 times as 2S1. The rest of the values in the first row re-

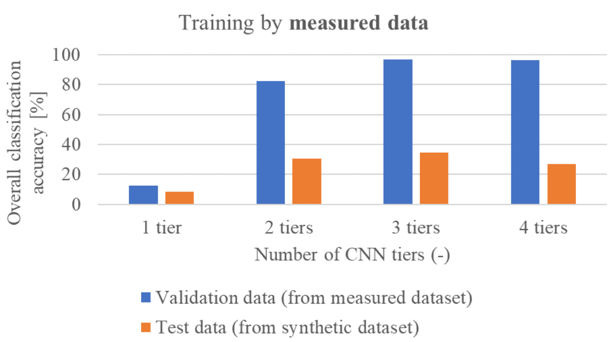


Fig. 2. The results of the CNNs trained by measured data.

2S1	**255**	0	0	0	5	1	0	3	0	0	96,6
BMP2	2	**145**	4	0	0	0	0	0	0	0	96,0
BTR70	0	1	**146**	0	1	2	0	0	1	0	96,7
M1	0	0	0	**215**	3	0	0	1	0	0	98,2
M2	0	0	0	4	**213**	0	0	0	0	0	98,2
M35	5	0	0	2	5	**203**	4	0	0	0	92,7
M548	0	0	0	0	0	1	**218**	0	0	0	99,5
M60	2	0	0	1	0	0	0	**259**	0	0	98,9
T72	4	1	0	6	2	0	0	0	**138**	0	91,4
ZSU23	0	0	0	0	0	0	2	2	0	**259**	98,5
	95,1	98,6	97,3	94,3	93,0	98,1	97,3	97,7	99,3	100,0	**96,9**
	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	

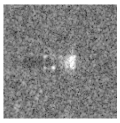
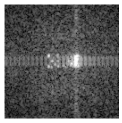
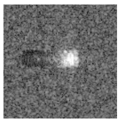
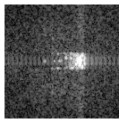
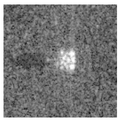
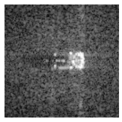
Fig. 3. Confusion matrix for **validation data** using CNN with 3 tiers (CNN trained by measured data).

presents incorrectly classified objects from a subset of images of the object 2S1: 5 times as M2, once as M35, and 3 times as M60. In the same way, we can analyze all objects from the confusion matrix.

In the last row and the last column, percentages of the number of classified objects for every particular class are shown. The column on the far right (the last column) shows the percentages of all the objects predicted to belong to each class that is correctly and incorrectly classified. We use given metrics to obtain the accuracy or positive predictive value. We can also compute the false discovery rate, to evaluate our presented approach. The row at the bottom of the confusion matrix shows the percentages of all the objects belonging to each class that is correctly and incorrectly classified. Computed values represent true positive rates, and we can also analyze false-negative rates. The cell in the bottom right of the plot shows the overall accuracy of the CNN with 3 tiers. The overall value of classification accuracy is satisfactory, we have reached the value of 96.9%. All objects from the input dataset are classified with an accuracy higher than 93% and only a few items from the validation data were misclassified. More important are the results of trained CNN achieved for test data (Fig. 4).

2S1	**83**	35	0	0	0	78	10	554	0	119	9,4
BMP2	14	**55**	0	0	0	19	2	239	0	173	11,0
BTR70	3	32	**0**	0	0	1	0	215	0	253	0,0
M1	29	14	5	**0**	0	4	7	543	3	124	0,0
M2	1	11	0	0	**0**	0	0	334	0	378	0,0
M35	4	0	0	0	0	**191**	367	10	0	157	26,2
M548	0	0	0	0	0	1	**703**	0	0	26	96,3
M60	80	1	2	0	0	24	2	**625**	0	140	71,5
T72	56	5	0	0	0	16	4	358	**0**	64	0,0
ZSU23	0	2	0	0	0	0	1	81	0	**792**	90,4
	30,7	35,5	0,0	0,0	0,0	57,2	64,1	21,1	0,0	35,6	**34,7**
	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	

Fig. 4. Confusion matrix for **test data** using CNN with 3 tiers (CNN trained by measured data).

Class and sensing angle values	Measured data	Synthetic data	View of real shapes
2S1 Elevation: 15° Azimuth: 180°			
M1 Elevation: 16° Azimuth: 180°			
M60 Elevation: 16° Azimuth: 180°			

Tab. 6. SAMPLE dataset SAR .png examples (128 × 128 pixels) and real shapes of 2S1, M1, and M60 objects.

The best results for test data of synthetic dataset were achieved by CNN with 3 tiers. Classification accuracy reached the value of 34.7% in this case. If we compare the results of the CNN with 3 tiers for the validation data with results for the test data of the synthetic dataset, the classification accuracy for the synthetic dataset is insufficient. The highest partial classification accuracy, about 60%, was reached for M548 and M35 object classes. For ZSU23, BMP2, and 2S1 object classes, a classification accuracy of around 30% was reached.

For the M60 object class, a classification accuracy of 21.1% was reached. For the rest of the object classes (BTR70, M1, M2, and T72), classification accuracy equals 0. Most misclassifications occurred in the 2S1 and M1 classes. Some objects were not correctly classified into the M60 class. These errors were most likely caused by the similarity of the geometric shapes of the vehicles that belong to the mentioned classes. Vehicles from classes 2S1, M1 and M60 are tracked and also equipped with a cannon. Views of real shapes of vehicles from classes 2S1, M1, and M60 are shown in Tab. 6.

Based on these results, hypothesis H1 can be rejected, and therefore it is not possible to use measured data to learn CNN and subsequently use learned CNN to recognize images of objects that have been created synthetically.

4.2 CNN Trained by Synthetic Data

To verify the second hypothesis, also four CNN topologies with one to four tiers (a complete structure is defined in Tab. 5) were created. Synthetic data (7050 images) were used for training. The input set was split at a usually used ratio of 70:30 [20], where 70 percent of the samples were used to form the training data. The remaining 30 percent was used to create validation data that was used to verify the quality of CNN's learning. A complete set of measured data containing 7050 images (Tab. 4) was used for testing. Results obtained by a given CNN topology with different tiers are shown in Fig. 5.

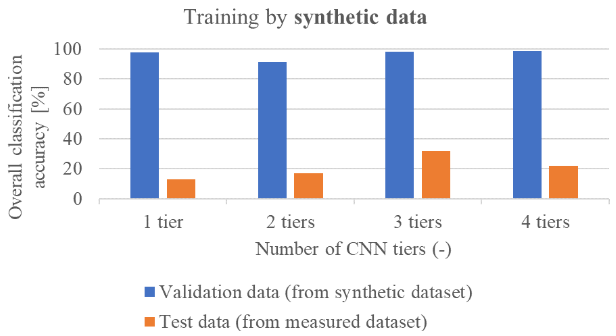


Fig. 5. The results of the CNNs trained by synthetic data.

Classification accuracy for the validation data is over 90% for all four CNNs. CNN with one tier reached a classification accuracy of 97%, CNN with two tiers at 92%, CNN with three tiers at 98.2%, and CNN with four tiers reached a classification accuracy of 98.7%. These results were promising but testing CNN with measured data turned out the very badly same way as in the previous case for hypothesis H1 was. Classification accuracy of CNN with one tier reached 12.9%, with two tiers 17%, with three tiers 31.7%, and with four tiers reached 21.8%.

The results of the CNN with 4 tiers for validation data are shown in Fig. 6, where we can find individual achievements for all object classes. For better results understanding, a confusion matrix was used. The meaning of particular items of the confusion matrix is the same as in the previous case for hypothesis H1. As an example, we will analyze one object class from the confusion matrix in detail. An object class BMP2 is presented in the second row, where the value in the second column represents the number of correctly classified images. It means that an object from class BMP2 was classified 146 times as BMP2. The rest of the values in the first row represents incorrectly classified objects from a subset of images of the class BMP2: 4 times as BTR70 and 2 times as 2S1. In the same way, we can analyze all objects from the confusion matrix.

The cell in the bottom right of the confusion matrix (Fig. 6) shows the overall classification accuracy of the CNN

2S1	**261**	3	0	0	0	0	0	0	0	0	98,9
BMP2	1	**146**	4	0	0	0	0	0	0	0	96,7
BTR70	1	0	**150**	0	0	0	0	0	0	0	99,3
M1	0	0	0	**211**	1	1	0	1	5	0	96,3
M2	0	0	1	0	**214**	0	0	0	0	2	98,6
M35	3	0	0	0	0	**216**	0	0	0	0	98,6
M548	0	0	0	0	0	1	**218**	0	0	0	99,5
M60	0	0	0	0	0	0	0	**261**	1	0	99,6
T72	0	1	0	0	0	0	0	2	**148**	0	98,0
ZSU23	0	0	0	0	0	0	0	0	0	**263**	100,0
	98,1	97,3	96,8	100,0	99,5	99,1	100,0	98,9	96,1	99,2	**98,7**
	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	

Fig. 6. Confusion matrix for **validation data** using CNN with 4 tiers (CNN trained by synthetic data).

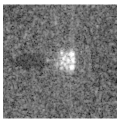
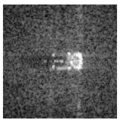
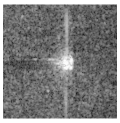
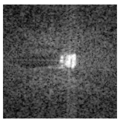
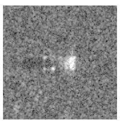
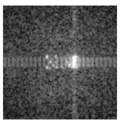
	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	
2S1	**405**	14	325	0	0	2	103	8	0	22	46,1
BMP2	52	**31**	378	0	0	0	10	4	0	27	6,2
BTR70	32	4	**464**	0	0	0	4	0	0	0	92,1
M1	36	30	77	**482**	9	0	10	63	0	22	66,1
M2	108	19	384	27	**8**	0	39	34	0	105	1,1
M35	129	29	200	74	0	**115**	154	25	0	3	15,8
M548	26	0	2	0	0	173	**529**	0	0	0	72,5
M60	437	28	3	176	1	0	15	**200**	0	14	22,9
T72	136	58	222	63	0	0	12	2	**0**	10	0,0
ZSU23	333	40	0	39	0	2	4	444	13	**1**	0,1
	23,9	12,3	22,6	56,0	44,4	39,4	60,1	25,6	0,0	0,5	**31,7**

Fig. 7. Confusion matrix for **test data** using CNN with 3 tiers (CNN trained by synthetic data).

CNN with 4 tiers. The overall value of classification accuracy is satisfactory, reaching a value of 98.7%.

All objects from the input data are classified with a probability higher than 96.1% and only a few items from the validation data were misclassified. More important are the results of trained CNN achieved for test data (Fig. 7).

The best results for test data of the measured dataset were achieved by CNN with 3 tiers. Classification accuracy reached the value of 31.7% in this case. Comparing results achieved by CNN with 3 tiers for validation data and results for test data (created from the measured dataset), classification accuracy for test data (created from measured data) is insufficient. The highest classification accuracy, around 60%, is observed for M548 and M1 object classes. Classification accuracy of around 20% is observed for object classes M60, BTR70, and 2S1. For object classes M2 and M35 we observed classification accuracy of around 40%. For BMP2 object class classification accuracy of 12.3% was achieved. Classification accuracy equal to 0 was achieved for object classes T72 and ZSU23. Most misclassifications occurred in M60 and ZSU23 classes.

Class and sensing angle values	**Measured data**	**Synthetic data**	**View of real shapes**
M60 Elevation: 16° Azimuth: 180°			
ZSU23 Elevation: 15° Azimuth: 180°			
2S1 Elevation: 15° Azimuth: 180°			

Tab. 7. SAMPLE dataset SAR .png examples (128 × 128 pixels) and real shapes of M60, ZSU23, and 2S1 objects.

Most of the objects belonging to these classes were not correctly classified to 2S1 respectively M60 class. These errors were most likely caused by the similarity of the geometric shapes of the vehicles that belong to the mentioned classes. Vehicles from classes M60, ZSU23, and 2S1 are tracked. View of real shapes of vehicles from classes M60, ZSU23, and 2S1 are shown in Tab. 7.

Based on these results, hypothesis H2 can be rejected, and therefore it is not possible to use synthetic data to learn CNN and subsequently use learned CNN to recognize images of objects that have been measured by real equipment.

4.3 CNN Trained by a Mixture of Measured and Synthetic Data

To verify the last – the third hypothesis, four CNN topologies with one to four tiers were developed again (a complete structure is defined in Tab. 5). For CNN training purposes, measured and synthetic datasets (14100 figures) were used. The input dataset was split at a usually used ratio of 70:15:15 [20], where 70 percent of figures were used to form the training data and 15 percent of figures were used to create validation data. The remaining subset (15 percent, containing 2115 images) was used for testing purposes. Results obtained by a given CNN topology with different tiers are shown in Fig. 8.

Classification accuracies of all four CNN structures are presented in Fig. 8. For validation data CNN with one tier reached a classification accuracy of 12.5%, CNN with two tiers at 90.2%, CNN with three tiers at 95.6%, and CNN with four tiers reached a classification accuracy of 97.3%. The results of the CNNs training are extremely good. Testing of the trained CNNs ended with very good results, which were opposite to both previous cases for hypotheses H1 and H2. For test data CNN with one tier reached a classification accuracy of 12.5%, CNN with two tiers at 88.5%, CNN with three tiers at 95.9%, and CNN with four tiers reached a classification accuracy of 96.6%.

The results of the CNN with 4 tiers for validation data are shown in Fig. 9, where we can find individual achievements for all object classes. For better results understanding, a confusion matrix was used again. The mean-

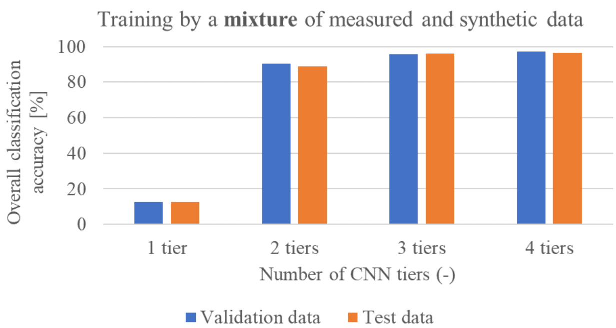


Fig. 8. The results of the CNNs trained by a mixture of measured and synthetic data.

	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	
2S1	**262**	0	0	0	0	1	0	1	0	0	99,2
BMP2	3	**139**	3	0	3	2	0	0	1	0	92,1
BTR70	3	7	**138**	0	3	0	0	0	0	0	91,4
M1	0	0	0	**215**	2	1	0	1	0	0	98,2
M2	0	0	0	1	**213**	2	0	1	0	0	98,2
M35	0	2	1	1	1	**210**	3	1	0	0	95,9
M548	0	0	0	0	0	0	**218**	0	0	1	99,5
M60	3	0	0	0	0	0	0	**259**	0	0	98,9
T72	0	2	0	1	2	0	0	2	**144**	0	95,4
ZSU23	0	0	0	0	8	0	1	7	0	**247**	93,9
	96,7	92,7	97,2	98,6	91,8	97,2	98,2	95,2	99,3	99,6	**96,6**

Fig. 9. Confusion matrix for **validation data** using CNN with 4 tiers (CNN trained by a mixture of measured and synthetic data).

	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23	
2S1	**252**	9	0	0	1	1	0	0	0	1	95,5
BMP2	2	**140**	3	0	1	0	0	0	4	1	92,7
BTR70	3	2	**144**	0	1	1	0	0	0	0	95,4
M1	0	0	0	**211**	1	0	0	4	3	0	96,3
M2	3	0	0	0	**211**	0	0	0	0	3	97,2
M35	0	2	0	1	3	**211**	2	0	0	0	96,3
M548	0	0	3	0	1	4	**211**	0	0	0	96,3
M60	1	0	0	4	0	0	0	**255**	2	0	97,3
T72	0	2	0	1	0	2	0	0	**146**	0	96,7
ZSU23	0	0	1	0	6	0	0	3	0	**253**	96,2
	96,6	90,3	95,4	97,2	93,8	96,3	99,1	97,3	94,2	98,1	96,1

Fig. 10. Confusion matrix for **test data** using CNN with 4 tiers (CNN trained by a mixture of measured and synthetic data).

ing of cells of the confusion matrix is the same as in the previous case for hypotheses H1 and H2. As an example, we will analyze one object class from Fig. 9 in detail. Results valid for object class BTR70 are presented in the third row, where the value in the third column represents the number of correctly classified images. It means that images containing object BTR70 were correctly classified 138 times as belonging to the BTR70 object class. The rest of the values in the third row represents incorrectly classified images containing object BTR70: 3 times as 2S1, 7 times as BMP2, and 3 times as M2. We can analyze all objects from the confusion matrix in the same way.

The cell in the bottom right of the confusion matrix (Fig. 9) shows the overall classification accuracy of the CNN with 4 tiers. The overall value of classification accuracy is satisfactory, reaching a value of 96.6%. All objects from the input data are classified with a probability higher than 91.8% and only a few items from the validation data were misclassified. More important are the results of trained CNN achieved for test data (Fig. 10). The best results for test data consisting of a mixture of the measured and synthetic datasets were achieved by CNN with 4 tiers. Classification accuracy reached a value of 96.1% in this case. Comparing results achieved by CNN with 4 tiers for validation data and results for test data, classification accuracies are almost the same. The highest classification accuracy, more than 98.1%, is observed for ZSU23 and M548 object classes. The lowest but still very good classification accuracy, around 90.3%, is observed for the BMP2 object class. For other object classes (2S1, BTR70, M2, M1, M35, M60, T72) are observed classification accuracy values from interval 93.8–97.3%. The worst results were achieved by CNN with 1 tier, which means that the given topology is not suitable for the classification of SAR images.

Based on these results, hypothesis H3 can be accepted, and therefore a mixture of measured and synthetic data can be used to learn CNN with very good results. Subsequently, CNN trained in that way can recognize images of objects that have been measured by real equipment and

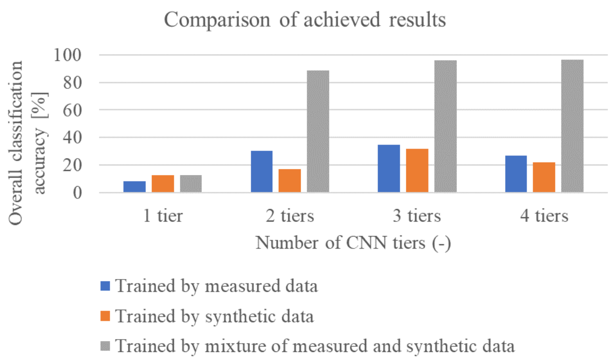


Fig. 11. Comparison of achieved results for different types of training set.

created in a synthetic environment. A visual comparison of all results achieved in CNN testing experiments is shown in Fig. 11.

Comparing achieved results with another SAMPLE dataset classification results [8] we can prove usability of proposed and tested CNN structures for SAMPLE dataset classification. We reached overall classification accuracy of 96.1%, authors in [8] declared, that they observed classification accuracy values from the interval of 88–95% for training data consisting of randomly selected objects from a set of measured (real) and synthetically created images of identical objects in a 1:1 ratio. Our achieved results also prove findings in [8] that it is not only important to match target configurations and sensor parameters, but it is also beneficial to match environmental conditions when generating synthetic SAR data for SAR image classification tasks. Environmental conditions, for example, wet snow, ground vegetation, hardwood species, and softwood species, have considerable effects on SAR imagery.

5. Conclusion

Hypotheses H1 (when CNN was trained only with measured data) and H2 (when CNN was trained only with synthetic data) can be rejected based on the results achieved by proposed CNNs. Thus, it is not possible to use real measured SAR images of objects to train CNN and then test and use the CNN with synthetic data. It is also not possible to do the opposite: train CNN with synthetic data and test and use CNN with real measured SAR images of objects. Hypothesis H3 (when CNN was trained with a mixture of measured and synthetic data) can be accepted as true because the learned CNN reached a classification accuracy for all objects close to 100%. The training data consisted of randomly selected objects from a set of measured (real) and synthetically created images of identical objects in a 1:1 ratio (the training set for each class contains 50% measured data and 50% synthetic data). In the same way, test data was created, i.e., a mix of randomly selected measured (real) and synthetic images that were not used in the CNN learning process. The obtained results confirm the suitability of the use of a mixture of measured and synthetic data for this purpose.

Authors in [7] and [8] just closely reached the classification accuracy of 95% for $k = 0.5$, i.e., the training set for each class contains 50% measured data and 50% synthetic data. Comparing achieved results with the above-mentioned SAMPLE dataset classification results we can prove the usability of proposed and tested CNN structures for automatic target classification of the synthetic aperture radar images. The classification accuracy of the original convolutional network is 96.1%, which is better than the results of the other research so far.

Acknowledgments

This research was funded by the Ministry of Defense of the Slovak Republic, grant number IM 4200517.

References

[1] SKOLNIK, M. I. *Radar Handbook.* 3rd edition. The McGraw-Hill Companies, 2008. ISBN: 978–0–07–148547–0

[2] MEIKLE, H. *Modern Radar Systems.* 2nd edition. Artech House, 2008. ISBN: 978–1–59693–242–5

[3] FERRO-FAMIL, L., POTTIER, E. Synthetic aperture radar imaging. *Microwave Remote Sensing of Land Surface*. December 2016, p. 1–65. DOI: 10.1016/B978–1–78548–159–8.50001–3

[4] AGER, T. P. An introduction to synthetic aperture radar imaging. *Oceanography*, 2013, vol. 26, no. 2, p. 20–33. DOI: 10.5670/oceanog.2013.28

[5] MASSONNET, D., SOUYRIS, J. C. *Imaging with Synthetic Aperture Radar*. 1st ed. EPFL Press, 2008. DOI: 10.1201/9781439808139

[6] SENSOR DATA MANAGEMENT SYSTEM (SDMS), SANDIA NATIONAL LABORATORIES. *MSTAR Overview (1995)*. [Online] Cited 2022–01–15. Available at: https://www.sdms.afrl.af.mil/index.php?collection=mstar

[7] LEWIS, B., SCARNATI, T., SUDKAMP, E., et al. A SAR dataset for ATR development: The Synthetic And Measured Paired Labeled Experiment (SAMPLE). In *Proceedings of SPIE Defense + Commercial Sensing Volume 10987, Algorithms for Synthetic Aperture Radar Imagery XXVI*; 109870H. Baltimore (Maryland, United States), 2019, p. 1–16. DOI: 10.1117/12.2523460

[8] SCARNATI, T., LEWIS, B. A deep learning approach to the Synthetic And Measured Paired Labeled Experiment (SAMPLE) challenge problem. In *Proceedings of SPIE Defense + Commercial Sensing Volume 10987, Algorithms for Synthetic Aperture Radar Imagery XXVI*; 109870H. Baltimore (Maryland, United States), 2019, p. 1–10. DOI: 10.1117/12.2523458

[9] SCARNATI, T., LEWIS, B. *Bullet Background Paper on Release of the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) Database to NATO* (28–January–2020)

[10] KUO, C. H., CHOU, Y. H., C., CHANG, P. C. Using deep convolutional neural networks for image retrieval. *Electronic Imaging*, 2016, vol. 28 p. 1–6. DOI: 10.2352/ISSN.2470–1173.2016.2.VIPC–231

[11] GAILLARD, M., EGYED-ZSIGMOND, E., GRANITZER, M. CNN features for Reverse Image Search. *Document numérique,* 2018/1–2, vol. 21, p. 63–90. DOI: 10.3166/DN.21.1–2.63–90

[12] AGNIHOTRI, A., SARAF, P., BAPNAD, K. R. A convolutional neural network approach towards self-driving cars. In *Proceedings of 2019 IEEE 16th India Council International Conference* (INDICON). Rajkot (India), 2019, p. 1–4. DOI: 10.1109/INDICON47234.2019.9030307

[13] OWAISALI CHISTI, S., RIAZ, S., BILALZAIB, M., et al. Self-driving cars using CNN and Q-learning. In *Proceedings of the 2018 IEEE 21st International Multi-Topic Conference (INMIC)*. Karachi (Pakistan), 2018, p. 1–7. DOI: 10.1109/INMIC.2018.8595684

[14] LAKHANI, P., SUNDARAM, B. Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology,* 2017, vol. 284, no. 2, p. 574–582. DOI: 10.1148/radiol.2017162326

[15] SARVAMANGALA, D. R., KULKARNI, R. V. Convolutional neural networks in medical image understanding: A survey. *Evolutionary Intelligence*, 2022, vol. 15, p. 1–22. DOI: 10.1007/s12065–020–00540–3

[16] KALASH, M., ROCHAN, M., MOHAMMED, N., et al. Malware classification with deep convolutional neural networks. In *Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS).* Paris (France), 2018, p. 1–5. DOI: 10.1109/NTMS.2018.8328749

[17] VESELÝ, J., OLIVOVÁ, J., GÖTTHANS, J., et al. Classification of microwave planar filters by deep learning. *Radioengineering*, 2022, vol. 31, no. 1, p. 69–76. DOI: 10.13164/re.2022.0069

[18] HUANG, G., LIU, Z., VAN DER MAATEN, L., et al. Densely connected convolutional networks. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition.* 2017, p. 1–9. DOI: 10.48550/arXiv.1608.06993

[19] HE, K., ZHANG, X., REN, S., et al. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016, p. 770–778. DOI: 10.48550/arXiv.1512.03385

[20] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*. MIT Press, 2016. ISBN: 9780262035613

About the Authors

Michal TURCANIK is an Associate Professor at the Department of Informatics at the Armed Forces Academy of general Milan Rastislav Štefánik in Liptovský Mikuláš. He has been teaching different courses for more than 20 years. He is an IST Panel Member of the NATO–STO organization for the Slovak Republic. His scientific research is focusing on reconfigurable logic, artificial intelligence, and computer networks.

Jozef PERDOCH is an Assistant Professor at the Department of Electronics at the Armed Forces Academy of general Milan Rastislav Štefánik in Liptovský Mikuláš. He successfully finished his dissertation studies in 2021. He is a SET Panel Member of the NATO–STO organization for the Slovak Republic. His research interests are focused on artificial intelligence applications, sensor systems, radar, and electronic warfare technologies.