HNNP – A Hybrid Neural Network Plait for Improving Image Classification with Additional Side Information

Ruth Janning, Carlotta Schatten and Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Hildesheim, Germany
Email: {janning, schatten, schmidt-thieme}@ismll.uni-hildesheim.de

***Abstract*—Most of the artificial intelligence and machine learning researches deal with big data today. However, there are still a lot of real world problems for which only small and noisy data sets exist. Hence, in this paper we focus on those small data sets of noisy images. Applying learning models to such data may not lead to the best possible results because of few and noisy training examples. We propose a hybrid neural network plait for improving the classification performance of state-of-the-art learning models applied to the images of such data sets. The improvement is reached by (1) using additionally to the images different further side information delivering different feature sets and requiring different learning models, (2) retraining all different learning models interactively within one common structure. The proposed hybrid neural network plait architecture reached in the experiments with 2 different data sets on average a classification performance improvement of 40% and 52% compared to a single convolutional neural network and 13% and 17% compared to a stacking ensemble method.**

***Keywords*-image classification; convolutional neural network; multilayer perceptron; hybrid neural network; side information; small data; noisy data;**

I. Introduction

Many problems treated in artificial intelligence and especially in machine learning nowadays concern big data like for instance data collected from social networks or consumer data. However, there are still also many real world problems delivering less data than desired for learning approaches and sometimes this data are additionally very noisy. In this paper we focus on those small noisy data sets, or on small data sets of noisy images respectively. Those data sets come for instance from problems in biomedicine, like cutaneous melanoma identification ([13]), brain tumor segmentation by magnetic resonance imaging ([9]) or analyzing microscopy images ([14]). Further examples are problems in the field of radar like detecting reflections of buried objects in ground penetrating radar images (see [6], [7], [8]) or recognizing volcanoes or craters on synthetic aperture radar images of the surface of planets like Mars or Venus ([15], [16], [19], [3]). The reason for small data sets in this fields is the often very expensive and time consuming recording of the images as well as the need of manual labeling. Most of those data are not publicly available. Furthermore, in many cases it is unavoidable that the recording method or the ambiance induce noise into the data. It suggests itself to use supervised learning methods for the classification task of the images in such data sets. However, applying supervised learning methods to small data sets usually delivers not the best possible classification performance. The reason is less broadness in training because of too few training examples in those data sets. Noise within the data is a further disturbing factor. Hence, the question arises, how to improve the classification performance of a learning model applied to such data. Very powerful state-of-the-art supervised learning tools for classifying images are convolutional neural networks (see e.g. [10], [18]). In this paper we will present an approach for improving the classification performance of a convolutional neural network as well as of ensemble methods applied to small data sets of noisy images. The performance improvement is reached by incorporating additional side information in two different ways: (1) using, additionally to the image pixel values, different further side information delivering different feature sets for the training and requiring different learning models, (2) retraining all different learning models interactively within one common structure with additional improved side information. This approach results in a plait of hybrid neural networks (fig. 1). We will demonstrate our approach by means of 2 different real data sets from the field of radar. In the experiments with this 2 data sets the proposed hybrid neural network plait (HNNP) was able to improve the classification performance by 40% and 52% compared to a convolutional neural network and by 13% and 17% compared to a stacking ensemble method (see sec. IV).

II. Related Work

The most famous problems convolutional neural networks are used for are handwritten digit classification and face recognition. Convolutional neural networks for face recognition and gender classification are presented e.g. in [12] and [21]. In [12] a rule-based algorithm for facial expression

recognition combined with face detection using a convolutional neural network is proposed for smile detection. The approach in [21] uses convolutional neural networks with a special type of processing element for gender classification. The well known LeNet-5 convolutional neural network for digits recognition is presented in [10]. A more recent convolutional neural network for digit recognition with a simpler and shallower architecture is proposed in [18]. The combination of neural networks to ensembles is investigated e.g. in [17] and an investigation of stacking, or stacked generalization, can be found e.g. in [20]. A combination of more than one convolutional neural network is proposed in [4], where a committee of 7 convolutional neural networks is used and the outputs are averaged. Different kinds of feature sets are used in [5], where 6 feature sets are trained by multilayer perceptrons. The outputs are merged by using another multilayer perceptron. The concept of using side information is also applied for clustering approaches ([23]) and distance metric learning methods ([22]). In [23] a distance metric using similarity information is learned and used for clustering with side information. The approach in [22] learns a Mahalanobis distance metric from side information for data clustering and classification. Our work combines the above mentioned approaches by using a committee of a convolutional neural network and multilayer perceptrons, different feature sets and additional side information. However, for our approach different kinds of neural networks with adapted architecture are retrained interactively within a plait structure using additional side information gained before and during the retraining for a further improvement.

III. The Hybrid Neural Network Plait

As mentioned above, the proposed hybrid neural network plait (HNNP) approach is based on two methods for incorporating additional side information: (1) integrating different information sources delivering different feature sets, (2) retraining the neural networks applied to this feature sets interactively. Point (1) implies that different neural networks are used for the different information sources. The different information sources are on the one hand the original information source – the pixel values of the image – and on the other hand sources of additional side information. The learning model for the original information source is, according to state-of-the-art approaches (sec. II), a convolutional neural network (sec. III-A). For the training of the feature sets of the other information sources simple fully connected multilayer perceptrons (sec. III-A) are used, as these feature sets are supposed to be less complex. Point (2) gives the plait the complex structure and incorporates in every retraining step improved additional side information. The plait (fig. 1) is composed of several layers in each of which the networks are retrained by considering the classification decisions of the other networks from the former layer (see sec. III-B).

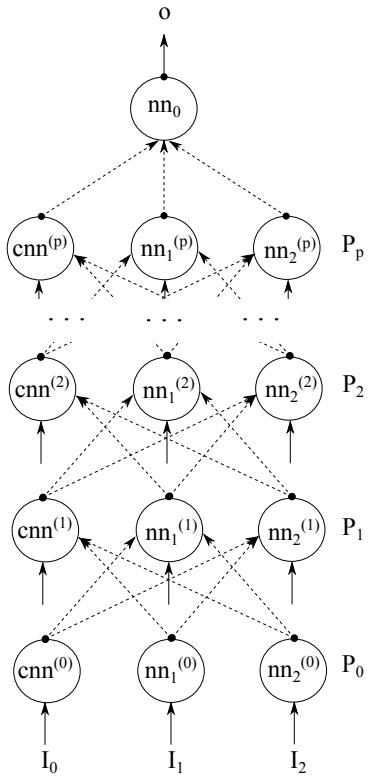

Figure 1. Architecture of the hybrid neural network plait (HNNP). The plait is composed of $p+1$ layers $P_0, P_1, \ldots, P_p$. Each layer contains $k+1$ different neural networks (here $k = 2$) according to the $k+1$ different information sources $I_0, I_1, \ldots, I_k$ for the input. I_0 corresponds to the input image and the appropriate learning model is a convolutional neural network (cnn). The learning models for the other information sources $I_1, \ldots, I_k$ are multilayer perceptrons ($nn_1, nn_2, \ldots, nn_k$). In every layer from P_1 on the neural networks are retrained with additional input from the former layer. After the last layer P_p a further multilayer perceptron (nn_0) is attached to archive one common output vector $\mathbf{o}$ delivering the final classification result.

A. Networks used for the Hybrid Neural Network Plait

Convolutional neural networks (cnn) like the LeNet-5 ([10]) are especially applicable for image classification. The original cnn consists of convolution layers, subsampling layers and fully connected layers. A convolution layer adapts the concept of *local receptive fields* by mapping overlapping areas of the original image to smaller neighboring areas of the feature maps in the following layer. The weights for the convolutions are shared, i.e. they are the same in every single feature map. A subsampling layer reduces the dimensionality by mapping several neurons of a feature map to one neuron of a smaller feature map in the next layer. These kinds of layers serve as a kind of local feature extractor. The fully connected layers after the convolution and subsampling layers on the other hand serve as a trainable classifier. By considering the concepts of *local receptive fields*, shared weights and spatial subsampling, cnns ensure to some degree shift, scale, and distortion invariance. In [18] a simplified cnn, which omits the subsampling layers and

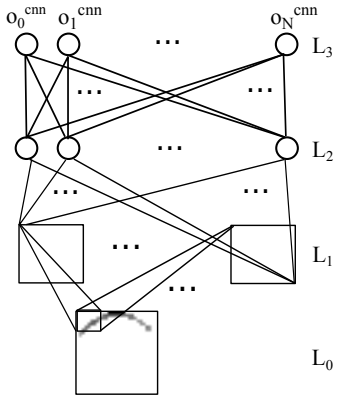

Figure 2. Architecture of a simplified convolutional neural network. It is composed of four layers L_0, L_1, L_2, L_3, where L_0 is the input layer fed with the image, L_1 is a convolution layer followed by a fully connected layer (L_2) and the output layer L_3. The output is the vector $(o_0^{cnn}, \ldots, o_N^{cnn})$ of the outputs of the neurons in the last layer.

uses only convolution layers, was proposed and evaluated. The cnns used for our approach are based on this architecture (fig. 2). Originally, there are 2 convolution layers, but for the sake of simplicity we pictured only 1 convolution layer in fig. 2. The pictured cnn consists of 4 layers L_0, L_1, L_2, L_3. L_0 is the input layer receiving the normalized pixel values of the input image. The next layer L_1 is a convolution layer followed by the fully connected layer L_2. The output layer L_3 delivers an output vector

$$\hat{\mathbf{o}}^{\mathbf{cnn}} := (\hat{o}_0^{cnn}, \ldots, \hat{o}_N^{cnn}), -1 \leq \hat{o}_i^{cnn} \leq 1, i = 0, \ldots, N \ , \tag{1}$$

where the actual output ${\hat{o}_i}^{cnn}$ of the ith neuron in output layer n corresponds to

$${\hat{o}_i}^{cnn} = \tanh(\sum_{l=0}^{N_{n-1}} w_n^{il} x_{n-1}^l) \ , \tag{2}$$

with N_{n-1} as the number of neurons in layer $n-1$, w_n^{il} as the weight of the lth connection between neuron i and the neurons in layer $n-1$, x_{n-1}^l as the output of the lth neuron in layer $n-1$ and $\tanh$ (hyperbolic tangent) as activation function with an output within the interval $[-1, 1]$. Each neuron of the output layer is assigned to one class in $\{C_0, \ldots, C_N\}$. The classification result $C_{\mathrm{argmax}_i \, {\hat{o}_i}^{cnn}}$ corresponds to the class C_i assigned to the output neuron with maximum output value ${\hat{o}_i}^{cnn}$. As usual in such architectures, the error in the last layer n for pattern $\mathcal{P}$ to be minimized is

$$E_n^{\mathcal{P}} := \frac{1}{2} \sum_{i=0}^{N} ({\hat{o}_i}^{cnn} - o_i^{cnn})^2 \ , \tag{3}$$

with ${\hat{o}_i}^{cnn}$ as actual output (see eq. (2)) and o_i^{cnn} as target output in

$$\mathbf{o}^{\mathbf{cnn}} := (o_0^{cnn}, \ldots, o_N^{cnn}), o_i^{cnn} \in \{-1, 1\}, i = 0, \ldots, N \ . \tag{4}$$

The training of this network for minimizing $E_n^{\mathcal{P}}$ for all patterns $\mathcal{P}$ is done by back-propagation, based on [11].

In opposite to the cnn, fully connected multilayer perceptrons are simpler structured and fulfill only a role as classifier, as they are equal to the last fully connected layers of a cnn. Such a fully connected multilayer perceptron uses the same formulas for $\hat{\mathbf{o}}^{\mathbf{nn_i}}, \mathbf{o}^{\mathbf{nn_i}}, E_n^{\mathcal{P}}$ (eq. (1),(4),(3)) as well as the same training method as the cnn.

B. Architecture of the Hybrid Neural Network Plait

The neural networks described in the section before are interweaved by the HNNP like in a plait (see fig. 1). This interweaving is enabled by adapting the architectures of the neural networks described above. In these new architectures (fig. 3) – using the example of the cnn – every neuron ${o_i^{cnn}}^{(m)}$ of the output layer L_n of a cnn in plait layer P_m is additionally connected to the outputs ${\hat{o}_i^{nn_1}}^{(m-1)}, {\hat{o}_i^{nn_2}}^{(m-1)}, \ldots, {\hat{o}_i^{nn_k}}^{(m-1)}$ of the k other networks $nn_1, nn_2, ..., nn_k$ from the previous plait layer P_{m-1}. This leads to new output formulas for the neural networks within the plait. In the case of k information sources for additional side information, the new output formulas for the $k+1$ adapted networks cnn and $nn_x, x = 1, \ldots, k$ are as follows:

$${\hat{o}_i^{cnn}}^{(m)} = \tanh(\sum_{l=0}^{N_{n-1}} w_n^{il} x_{n-1}^l + \sum_{j=1}^{k} b_j^i {\hat{o}_i^{nn_j}}^{(m-1)}) \ , \tag{5}$$

$$\begin{aligned} {\hat{o}_i^{nn_x}}^{(m)} = \tanh(&\sum_{l=0}^{N_{n-1}} w_n^{il} x_{n-1}^l + b_0^i {\hat{o}_i^{cnn}}^{(m-1)} \\ &+ \sum_{j=1, j \neq x}^{k} b_j^i {\hat{o}_i^{nn_j}}^{(m-1)}) \ . \end{aligned} \tag{6}$$

In each of the formulas there are k new additional weights b_j^i to learn. That means that each neural network learns how to consider the decisions of all other networks from the plait layer before. With other words, each network learns to which degree it may trust the former classification of each of the other networks. The number $p+1$ of plait layers is a hyper parameter. It can easily be estimated heuristically as it corresponds to the number of sequential layers which decrease the classification error during the training phase. If the classification error no longer decreases, the plait construction is stopped. In our experiments in sec. IV for instance p had the value 2, i.e. the HNNP had 3 layers. As the training and testing within one plait layer can be parallelized, the time complexities $t_{train}(\mathrm{HNNP})$ and $t_{test}(\mathrm{HNNP})$ for training and testing of the HNNP are proportional to the training and testing time complexities of the one neural network nn^{max} within the plait which has the highest time complexity for training and testing. That is if $t_{train}(nn^{max})$ and $t_{test}(nn^{max})$ are the time complexities for training and for testing of nn^{max}, then the time complexities

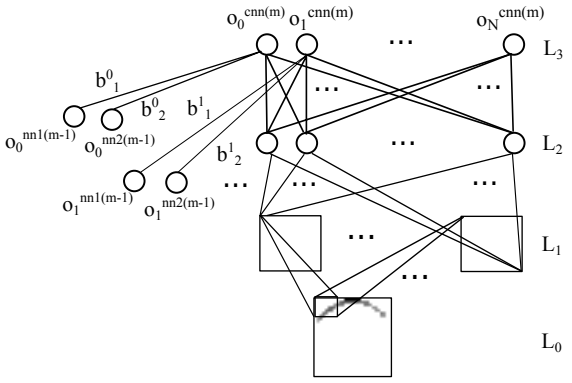

Figure 3. Architecture of the cnn of fig. 2 adapted to the plait structure with 3 information sources. In contrast to fig. 2 the output neurons of this cnn get additional input from the other networks of the plait layer before.

for training and testing of the HNNP are

$$t_{train}(\text{HNNP}) = (p+1) \cdot t_{train}(nn^{max}) \ , \tag{7}$$

$$t_{test}(\text{HNNP}) = (p+1) \cdot t_{test}(nn^{max}) \ . \tag{8}$$

The final component of the HNNP is an additional subsequent fully connected multilayer perceptron nn_0 (see fig. 1). nn_0 is fed with the outputs of every neural network in the last plait layer P_p and delivers one common final classification decision. How to consider the single classification decisions of the single neural networks, to come to one common classification decision, is learned by nn_0 similarly to the stacking approach. The difference to simple stacking is that HNNP retrains the involved neural networks within several layers with additional improved side information.

IV. Experiments

To evaluate our HNNP approach, we applied it to 2 different real data sets. The first data set (sec. IV-A) is composed of ground penetrating radar (GPR) images. The radargrams where recorded on a real test field in Germany. The resulting GPR images[1] are recordings of certain areas of the subsurface and contain hyperbola shaped reflections belonging to buried objects. The goal is to detect buried pipes by recognizing the corresponding reflections. The second data set (sec. IV-B) comes from a set of synthetic aperture radar (SAR) images of the surface of Venus, available at the UCI Machine Learning Repository [1]. The data were collected by the Magellan spacecraft from 1990–1994 to obtain a mapping of the surface of Venus by means of SAR technology ([15], [16]). The classification task in this data is the identification of volcanoes. For both data sets we used the same experimental settings: our HNNP approach is compared to the five baselines cnn, nn_1, nn_2, *majority ensemble* and *stacking ensemble*. cnn is a single convolutional neural network (sec. III-A) fed with the normalized pixel values of the images to classify. nn_1, nn_2 are single fully connected multilayer perceptrons (sec. III-A) with input feature sets coming from the appropriate additional side information. *Majority ensemble* classifies according to the majority vote of cnn, nn_1 and nn_2. *Stacking ensemble* learns to combine the classification decisions of cnn, nn_1 and nn_2 by using the subsequent multilayer perceptron nn_0. Furthermore, we chose a data split of 5 for a 5-fold cross validation and hence carried out five different experiments with every data set.

[1]Generated for the EFRE project AcoGPR (http://acogpr.ismll.de).

A. The Ground Penetrating Radar Data Set

The GPR data set consists of 10 radargrams. From these radargrams we got 2707 manually labeled examples – gained by a modification of a clustering approach (GamRec, [6], [8]) for recognizing hyperbola shaped areas within an image – composed of 1166 hyperbola and 1541 noise examples. In the archived example set we identified 5 different hyperbola and 5 different noise types with different shapes. Hence, every example is assigned to one of 10 classes to respect the different shape types (see fig. 4). The cnn, nn_1, nn_2 and *majority ensemble* were trained with this 10 classes in our experiments. The 2 main classes (hyperbola, noise) were considered only in the end by counting examples assigned to class $0, \ldots, 4$ as hyperbolas and examples assigned to class $5, \ldots, 9$ as noise. For *stacking ensemble* and HNNP also every network was trained with the 10 classes except the subsequent multilayer perceptron nn_0. nn_0 distincts only between the 2 main classes. Although the final results are binary, the 10 subclasses are needed for training to reduce the large intra-class variance within the 2 main classes, as it cannot be covered by the small amount of train examples. The average sizes of train and test set for the 5 splits are 2166 and 541. Additionally to the images we have chosen 2 further information sources: statistical information (input for nn_1) and domain specific information (input for nn_2). The statistical information features are on the one hand the numbers of cluster pixels occurring in one of 16 areas of the cluster image (partitioned by a 4×4 grid) and on the other hand the numbers of cluster pixels with 0, 1, 2, ... or 8 neighbors, resulting in 25 features. The domain specific information delivers 8 features concerning the GPR image and the original clusters found in this image.

B. The Synthetic Aperture Radar Data Set

For the SAR data set we have chosen 5 images of the Magellan data set. We extracted by means of a further modification of GamRec without the hyperbola shape checking 898 examples consisting of 451 volcano and 447 noise examples, assigned to one of 10 subclasses (see fig. 4). The average sizes of train and test set for the 5 splits are 718 and 180. The input feature sets for nn_1 and nn_2 come from statistical information. nn_1 is fed with the histogram of the input image. nn_2 is fed with the number of certain pixels occurring in one of 100 areas of the image (partitioned by a

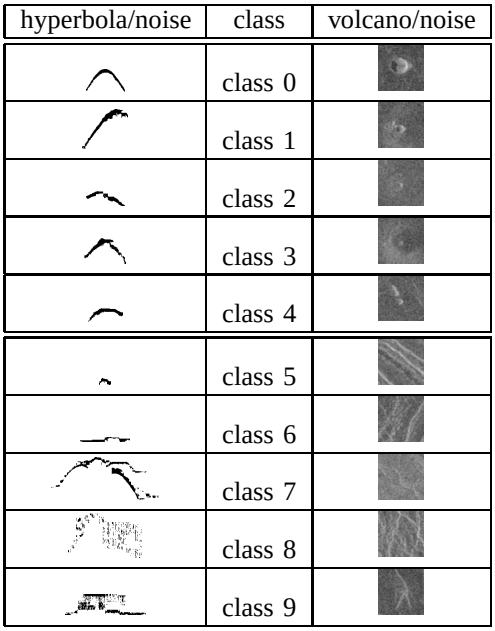

Figure 4. Examples of the 10 hyperbola/volcano and noise classes of the GPR (left) and the SAR data set (right).

Table I
RESULTS FOR THE GPR DATA SET: CLASSIFICATION TEST ERRORS (%) OF THE 5 BASELINES cnn, nn_1, nn_2, *majority ensemble* AND *stacking ensemble* AND THE HNNP. THE RESULTS FOR EACH OF THE 5 DATA SPLITS FOR THE CROSS VALIDATION ARE SHOWN IN THE FIRST 5 ROWS, THE AVERAGE RESULT (AVRG.) IS GIVEN IN THE LAST ROW (STANDARD DEVIATIONS IN BRACKETS).

split	cnn	nn_1	nn_2	majority	stacking	HNNP
1	17.84	16.39	32.16	14.94	14.52	**12.66**
2	15.04	13.45	31.33	12.92	12.57	**11.50**
3	22.80	14.17	31.11	14.33	11.40	**10.42**
4	19.65	13.10	32.57	14.16	10.80	**9.38**
5	18.09	16.22	31.39	13.31	14.55	**11.64**
avrg.	18.68 (2.84)	14.67 (1.55)	31.71 (0.62)	13.93 (0.81)	12.77 (1.73)	**11.12** (1.26)

10×10 grid). The considered pixels must have a gray value within the upper quarter of all gray values.

C. Results

The results of the experiments with the GPR and the SAR data set are shown in table I and table II. In table III the percentaged improvement of the classification error by our HNNP in comparison to the cnn and the *stacking ensemble* is presented. As one can see in table I and II, for each split of the data sets for the cross validation and on average, our proposed HNNP approach was able to reduce the classification error in comparison to the classification error of a single convolutional neural network and of the other networks as well as in comparison to the classification error of both ensemble methods. One could suppose that the ensemble

Table II
RESULTS FOR THE SAR DATA SET: CLASSIFICATION TEST ERRORS (%) OF THE 5 BASELINES cnn, nn_1, nn_2, *majority ensemble* AND *stacking ensemble* AND THE HNNP. THE RESULTS FOR EACH OF THE 5 DATA SPLITS FOR THE CROSS VALIDATION ARE SHOWN IN THE FIRST 5 ROWS, THE AVERAGE RESULT (AVRG.) IS GIVEN IN THE LAST ROW (STANDARD DEVIATIONS IN BRACKETS).

split	cnn	nn_1	nn_2	majority	stacking	HNNP
1	21.23	20.11	16.20	17.88	15.64	**14.53**
2	30.56	22.22	15.56	17.78	12.78	**11.11**
3	37.43	34.64	21.23	28.49	20.67	**16.76**
4	39.44	38.89	27.78	33.33	25.56	**18.33**
5	30.00	27.78	20.00	25.56	16.67	**15.00**
avrg.	31.73 (7.19)	28.73 (8.00)	20.15 (4.90)	24.61 (6.78)	18.26 (4.96)	**15.15** (2.71)

Table III
IMPROVEMENTS (%) OF THE CLASSIFICATION TEST ERROR BY HNNP IN COMPARISON TO THE cnn AND THE *stacking ensemble*.

split	% error reduction compared to stacking		% error reduction compared to cnn	
	GPR	SAR	GPR	SAR
1	12.81	7.10	29.04	31.56
2	8.51	13.07	23.54	63.65
3	8.60	18.92	54.30	55.22
4	13.15	28.29	52.26	53.52
5	20.00	10.02	35.66	50.00
avrg.	12.92	17.07	40.47	52.27

methods perform better than the single neural networks, however the *majority ensemble* performs even worse in some cases. The reason might be that the *majority ensemble* interprets the classifications of the involved networks directly and cannot cover the cases in which only one network delivers the right classification decision. The *stacking ensemble* in contrast learns how – with which weighting – to interpret the outputs of the involved networks. Hence, in table III we compared HNNP only with the state-of-the-art neural network cnn and the stronger *stacking ensemble*. Table III shows that on average HNNP reached an error reduction of approximately 40% and 52% in comparison to the cnn applied to the GPR and the SAR data set and an error reduction of approximately 13% and 17% in comparison to the *stacking ensemble*. HNNP performs better than the *stacking ensemble*, as stacking corresponds to only one layer of networks whereas HNNP contains several network layers for retraining, in which also the single networks learn how to consider the classification decisions of the other networks. The networks within the plait perform better after retraining and commit their improved classification result to the networks in the following layer. In this way, the classification error is more and more improved over the plait layers until the classification skills of the networks are

exhaustively merged. The results of our experiments show that for the considered data sets *stacking ensemble* is not sufficient and a retraining in plait layers delivers further classification improvements. In summary, we compared our approach to five baselines, each of which was outperformed by HNNP. Especially in comparison to the state-of-the-art approch cnn and the stronger ensemble method *stacking* HNNP reached a significant error reduction.

V. Conclusion

In this paper we addressed the problem of applying learning models for classification to small noisy data sets. We presented a hybrid neural network plait architecture (HNNP), which is able to improve the classification performance in such data sets in comparison to applying a single convolutional neural network or ensemble methods to them.

Acknowledgment

This work is co-funded by the EFRE project AcoGPR.

References

[1] K. Bache and M. Lichman, *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]*, Irvine, CA: University of California, School of Information and Computer Science, 2013.

[2] S. Birkenfeld, *Automatic detection of reflexion hyperbolas in gpr data with neural networks*, World Automation Congress (WAC), pp. 1–6, 2010.

[3] M. C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler and J. Aubele, *Learning to Recognize Volcanoes on Venus*, Machine Learning, 1998.

[4] D. C. Ciresan, U. Meier, L. M. Gambardella and J. Schmidhuber, *Convolutional Neural Network Committees For Handwritten Character Classification*, International Conference on Document Analysis and Recognition, 2011.

[5] R. M. O. Cruz, G. D. C. Cavalcanti and T. I. Ren, *Handwritten Digit Recognition Using Multiple Feature Extraction Techniques and Classifier Ensemble*, 17th International Conference on Systems, Signals and Image Processing, 2010.

[6] R. Janning, T. Horváth, A. Busche and L. Schmidt-Thieme, *GamRec: a Clustering Method Using Geometrical Background Knowledge for GPR Data Preprocessing*, Artificial Intelligence Applications and Innovations (IFIP Advances in Information and Communication Technology 381), pp. 347–356, Springer, Heidelberg, 2012.

[7] R. Janning, T. Horváth, A. Busche and L. Schmidt-Thieme, *Pipe Localization by Apex Detection*, Proceedings of the IET international conference on radar systems (Radar 2012), Glasgow, Scotland, 2012.

[8] R. Janning, A. Busche, T. Horváth and L. Schmidt-Thieme, *Buried Pipe Localization Using an Iterative Geometric Clustering on GPR Data*, Artificial Intelligence Review, DOI: 10.1007/s10462-013-9410-2, Springer, 2013.

[9] V. G. Kanas, E. I. Zacharaki, E. Dermatas, A. Bezerianos, K. Sgarbas and C. Davatzikos, *Combining Outlier Detection with Random Walker for Automatic Brain Tumor Segmentation*, Artificial Intelligence Applications and Innovations (IFIP Advances in Information and Communication Technology 382), pp. 26–35, Springer, Heidelberg, 2012.

[10] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-Based Learning Applied to Document Recognition*, Proceedings of the IEEE 86 (11), pp. 2278–2324, 1998.

[11] Y. LeCun, L. Bottou, G. Orr and K. Muller, *Efficient BackProp*, Neural Networks: Tricks of the trade, 1998.

[12] M. Matsugu, K. Mori, Y. Mitari and Y. Kaneda, *Subject independent facial expression recognition with robust face detection using a convolutional neural network*, Neural Networks 16, pp. 555–559, Elsevier, 2003.

[13] K. Moutselos, A. Chatziioannou and I. Maglogiannis, *Feature Selection Study on Separate Multi-modal Datasets: Application on Cutaneous Melanoma*, Artificial Intelligence Applications and Innovations (IFIP Advances in Information and Communication Technology 382), pp. 36–45, Springer, Heidelberg, 2012.

[14] P. A. Nogueira and L. F. Teófilo, *A Probabilistic Approach to Organic Component Detection in Leishmania In fected Microscopy Images*, Artificial Intelligence Applications and Innovations (IFIP Advances in Information and Communication Technology 381), pp. 1–10, Springer, Heidelberg, 2012.

[15] G. H. Pettengill, P. G. Ford, W. T. K. Johnson, R. K. Raney and L. A. Soderblom, *Magellan: Radar Performance and Data Products*, Science, 252:260–265, 1991.

[16] R. S. Saunders, A. J. Spear, P. C. Allin, R. S. Austin, A. L. Berman, R. C. Chandlee, J. Clark, A. V. Decharon and E. M. Dejong, *Magellan Mission Summary*, Journal of Geophysical Research Planets, 97(E8):13067–13090, 1992.

[17] A. J. C. Sharkey and N. E. Sharkey, *Combining diverse neural nets*, The Knowledge Engineering Review, Vol. 12:3, pp. 231–247, 1997.

[18] P. Y. Simard, D. Steinkraus and J. Platt, *Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis*, External Link International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, Los Alamitos, pp. 958–962, 2003.

[19] P. Smyth, M. C. Burl, U. M. Fayyad and P. Perona, *Knowledge Discovery in Large Image Databases: Dealing with Uncertainties in Ground Truth*, In: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Menlo Park, CA, 1995.

[20] K. M. Ting and I. H. Witten, *Issues in stacked generalization*, Journal of artificial intelligence research, Vol. 10, pp. 271–289, 1999.

[21] F. H. C. Tivive and A. Bouzerdoum, *A Shunting Inhibitory Convolutional Neural Network for Gender Classification*, 18th International Conference on Pattern Recognition 2006 (ICPR 2006), IEEE, pp. 421–424, 2006.

[22] S. Xiang, F. Nie and C. Zhang, *Learning a Mahalanobis distance metric for data clustering and classification*, Pattern Recognition Vol. 41, Issue 12, pp. 3600–3612, Elsevier, 2008.

[23] E. P. Xing, A. Y. Ng, M. I. Jordan and S. Russell, *Distance Metric Learning, with Application to Clustering with Side-Information*, Advances in Neural Information Processing Systems 15, pp. 505–512, 2002.