Radial Basis Function Neural Network with a Centers Training Stage for Prediction Based on Dispersed Image Data

Kwabena Frimpong Marfo[1][0000−0003−2226−9097] and Małgorzata Przybyła-Kasperek[1][0000−0003−0616−9694]

University of Silesia in Katowice, Institute of Computer Science, Będzińska 39, 41-200 Sosnowiec, Poland
{kwabena.marfo,malgorzata.przybyla-kasperek}@us.edu.pl

Abstract. Neural networks perform very well on difficult problems such as image or speech recognition as well as machine text translation. Classification based on fragmented and dispersed data representing certain properties of images or computer's vision is a complex problem. Here, the suitability of a Radial Basis Function (RBF) neural network was evaluated using fragmented data in the problem of recognizing objects in images. The great difficulty of the considered problem is, there is not images data as such but only data on some properties of images stored in a dispersed form. More specifically, it was demonstrated that applying a $k-$nearest neighbors classifier in the first step to generate predictions based on fragmented data, and then using a RBF neural network to learn how to correctly recognize the systems of generated predictions for making a final classification is a good approach for recognizing objects in images. An additional step of training the weights (centers) between the input and hidden layers of a RBF network was proposed. In general, this investigation demonstrates that adding this step significantly improves the correctness of recognizing objects in images.

Keywords: Radial Basis Function Neural Network · Dispersed Data · Image Data Processing.

1 Introduction

Object detection in images is very important in today's world and many applications such as surveillance systems can be found. Examples of important uses include: recognizing types of vehicles in road traffic [18], recognizing types of objects in satellite images [12], or even recognizing components on a production line [20, 2]. In literature, we can find numerous applications of neural networks for processing images and recognizing objects in images [21, 19, 1]. These are mainly applications of convolutional neural networks. A very interesting approach where neural networks were used for recognizing handwriting can be found in [14]. In [5], the generative adversarial network (GAN) was used to generate images. Paper [15] provided an overview of very interesting approaches that used neural networks to generate artistic patterns.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

It is very rare to find studies that deal with recognizing images that are available in fragmentary form. This problem can be considered as a set of images obtained from the perspective of several cameras, each of which observes the object from different angles [16]. In such a case – when the image data is fragmented, recognizing the object in the photo is more difficult. In the paper [7], the approach of assembling fragments of photos and matching parts to merge into one can be found. This approach required that the fragmented data do not overlap, however, in this study, fragmentation concerns the dispersion of information about the recognized object with shared fragments and does not mean disjointed information. We also assume that we do not have images as such, nor its fragments, but only the characteristics extracted from these images. These may be the average values recorded in a certain area of pixels or certain shape characteristics such as the length and width depicted in the image. We also assume that these data are available in tabular form – a set of decision tables. However, the data may overlap, i.e. there may be common conditional attributes in several decision tables. In addition, in the training set which comprises a set of local tables, there may appear fragmentary characteristics for the same physical object or for other objects belonging to the same decision class. The focus of this study was not on image fusion but on object recognition, assigning the correct decision class for an object that is seen in a fragmented way.

Other approaches used to recognize fragmented images can also be found in literature. The paper [22] proposed the use of hidden Markov models for gesture recognition based on fragmentary vision. In the paper [3], fuzzy rules were used for fragmented handwritten digit recognition.

This paper proposes the use of a Radial Basis Function (RBF) neural network with a centers training stage in combination with the $k-$nearest neighbors algorithm to classify objects observed in images based on fragmented characteristics – data stored as a set of local decision tables. For this purpose, three problems were considered: classification of car type based on photo's characteristics, classification of land type based on satellite images, and classification of bean type based on fragmentary computer vision. To the best of our knowledge, such an issue has not been studied before in literature. In this study research results, comparisons and statistical tests are presented. It has been justified that the proposed approach gives much better results than the RBF networks without a centers training stage as well as the baseline approach that uses a heterogeneous ensemble of classifiers.

The paper is organized as follows. In Section 2, the proposed classification model using a RBF neural network is described. The algorithm's description and the discussion about the key features of the proposed approach are given. Section 3 addresses the datasets that were used and presents the conducted experiments and discussion on obtained results. Section 4 is on conclusions and future research plans.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

2 Model and methods

When data is stored in dispersed form, aggregating local tables into a table becomes a difficult task due to data inconsistencies. To overcome this, we consider local data separately, look for patterns in them and find a way to combine the dependencies already discovered into a single piece. More formally, we assume that some characteristics of images are available in dispersed form – in the form of a set of local tables. We assume that a set of decision tables $D_i = (U_i, A_i, d)$, $i \in \{1, \ldots, n\}$ is available, where U_i is the universe comprising a set of objects – images; A_i is a set of conditional attributes – some features that describe the image; d is a decision attribute – object shown in the image. Objects and attributes in local tables can be different, however, some objects may be common among local tables as we do not impose any restrictions here.

This is a case where different sensors capture features of an image based on the object identified in the image. In a real-life situation, it could be pictures taken at different angles of the same object by different cameras. Thus, local tables will be generated (one local table for each camera) with different sets of conditional attributes (but some may be shared). Another instance could be cameras set up in different locations in a city where each camera takes pictures of vehicles on the road and based on the features identified, the type of vehicle is determined. In this situation, same object could be recognized by different cameras, but most of the objects identified will be different.

Since aggregation of these local tables into a single table is not immediately possible, one possible approach that can be used is to generate classifiers based on each local table separately. For this purpose, the $k-$nearest neighbors classifier is chosen, as it has low computational complexity and is a suitable method for image classification based on local features [8]. We expect objects of one type in images to have similar features. To calculate the distances between the test objects and the objects in the local tables, the Gower measure is used [13]. This measure allows to compute the distance even when there are attributes of different types (quantitative, qualitative, binary) and from different ranges in the decision table (it does not require normalization or standardization). In addition, it should be noted that for the test objects, values of all attributes occurring in the local tables should be known. Each base classifier makes its classification using a subset of all attributes – more strictly, a classifier i that is built based on a decision table D_i uses the set of attributes A_i. A modification of the $k-$nearest neighbors classifier is proposed, i.e. instead of generating a prediction from the abstract level (a decision class most frequent in the neighborhood), a prediction from the measurement level is generated. That is, a classifier i generates a probability vector over decision classes for a test object x (denoted by $\mu_i(x)$). The dimension of vector $\mu_i(x) = [\mu_{i,1}(x), \ldots, \mu_{i,c}(x)]$ is equal to the number of decision classes $c = card\{V^d\}$, where V^d is a set of decision attribute values (a set of the types of objects in the image), $card\{V^d\}$ is the cardinality of this set. Each coefficient $\mu_{i,j}(x)$ is determined using the k-nearest neighbors of the test object x belonging to a given decision class j and decision table D_i. In this way, the information we get from the base classifiers is more complete. These

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

are average similarities determined from fragmented data for each decision class (i.e. objects that may appear in the image).

To summarize the previous step, the base classifiers for the test object x generates n prediction vectors $\mu_i(x)$, $i \in \{1, \ldots, n\}$, each vector is c dimensional, i.e. a composite of similarities to decision classes obtained based on fragmented data. The task of recognizing the correct decision class based on such vectors – identifying the object in the image – is a difficult task. For this purpose the Radial Basis Function (RBF) neural network is used.

We make use of a RBF neural network because they have the property of separating spaces that are not linearly separable. This is achieved by transforming the input vectors into a new feature space, where classes are linearly separable. For this to be possible, there must be more neurons in the hidden layer than in the input layer. In the considered problem, the input vector is formed through the prediction vectors generated by the base classifiers. So, more formally, a vector will be created $[\mu_1(x), \ldots, \mu_n(x)]$ for the test object x. Its dimension is equal to $n \cdot c$ because we have n base classifiers, each generating a c dimensional vector. Thus, we have $n \cdot c$ neurons in the input layer. RBF neural networks always contain only one hidden layer, thus, there are no problems with determining the appropriate number of hidden layers for a given problem. Each neuron in the hidden layer has a parameter called center. Formally, the $k-$th neuron in the hidden layer has the center c_k. The center is interpreted as a representative of a certain group of objects. RBF networks are similar to the $k-$nearest neighbors approach. However, instead of eliminating distant objects from the classification process (as it is done in the $k-$nearest neighbors classifier), this time we simply reduce the influence of distant objects on the output of the neural network, but still use them in the classification process. We obtain this property by using the Gaussian function as the activation function for each of the neurons in the hidden layer. The more the input vector is similar to the center of the neuron in the hidden layer, the greater its influence on the output of the neural network. Let us denote the vector given as the input of the network in the input layer by y. Then for the $k-$th neuron of the hidden layer the Gaussian function is as follows

$$\Phi_k(y) = \exp\left[-\frac{\| y - c_k \|}{2\sigma_k^2}\right], \tag{1}$$

where c_k is the center of the $i-$th neuron, σ_k is the $k-$th neuron's bandwidth and $\| \cdot \|$ is the Euclidean norm. The Gaussian width σ_k of the $k-$th neuron in the hidden layer was estimated as $\sigma_k = \frac{\rho \max}{2 \cdot n_H}$ where $\rho_{\max}$ is the maximum distance between the chosen centers and n_H is the number of neurons in the hidden layer. The weights and biases assigned to the connections of neurons from the hidden layer to the output layer in the RBF network are trained using the back-propagation algorithm.

A very important issue in RBF networks is the appropriate determination of the center of neurons in the hidden layer. This is usually done by a clustering algorithm realized on the training set (this is implemented before the training of the network). Usually the $k-$means clustering algorithm is used and the centroids

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

determined by this algorithm are used as centers connecting the input layer to the hidden layer neurons. The Lloyd's $k-$means algorithm – a modification of the $k-$means algorithm is also very often used. In the study, one of the approaches analyzed is the Lloyd's $k-$means algorithm for determining centers. But more interesting is the proposed approach in which instead of designating centers only once (determined by the Lloyd's $k-$means algorithm), a step of training these centers is used. This is implemented as follows.

- In the first stage, the Lloyd's $k-$means algorithm is used to determine the centers which serves as connections between the input layer and the hidden layer.
- After, random values are assigned as weights and biases, which serve as connections between the hidden layer and the output layer.
- Then, training the RBF network begins, however, here the propagation of the error is extended up to the centers. This way, the centers are trained iteratively together with the weights and biases using the back-propagation method.

The pseudo-code for the proposed RBF neural network is given below in Listing 2.

RBF Neural Network with a Centers Training Stage

```
# X: matrix of prediction vectors over all local tables
# centers: array of values determined by the Lloyd's algorithm
# neurons: number of neurons in the hidden layer
# sigma: Gaussian width

def RBFModel(X:array, centers:array, neurons:int, sigma:float):
  model = Sequential()
  model.add(Flatten(input_shape=(X.shape[1],)))
  model.add(RBFLayer(units= neurons, gamma=sigma))
  model.add(Dense(classes, activation='softmax'))
  model.layers[1].set_weights([centers])
  model.compile(loss='categorical_crossentropy',
  optimizer='adam',metrics=['accuracy'])
  return model
```

The above describes how to build and train the RBF network. An important issue still is the set used for training, the number of epochs used and the optimal batch-size. In the problem presented here, local tables were used to generate prediction vectors for objects. In addition, the local tables contain only fragmented data, and to train the network we need objects that have values for all attributes that are present in the local tables (in order to designate vectors for all tables). So, to train the RBF network, a stratified 10-fold cross-validation method on the test set was used. That is, the test set was divided into 10 folds with equal number of objects and proportional shares of decision classes. At

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

each iteration, the RBF network was trained using 9 folds (to be very accurate – prediction vectors generated for test objects from these 9 folds), and the quality of classification of the model was evaluated on the last independent fold. This procedure was repeated three times and the results was averaged to determine the classification error level of the model. To determine the number of epochs and batch-size, different values were experimented to determine the optimal values for each dataset. After numerous trials, the optimal (epoch, batch-size) for Vehicle Silhouettes, Landsat Satellite and Dry Bean datasets were (400, 200), (400, 500), (400,15) respectively. RBF networks trains considerable fast so the whole process was fairly quick. A big advantage of the RBF networks was how interpretable the results obtained from it were. This was achieved thanks to properly selected/trained centers and their reduced influence in the case of large distances.

3 Datasets and results

The system proposed in the paper for the classification of objects in images based on characteristics stored in fragmented form – a set of local decision tables – was tested on three datasets. These datasets in their original form were retrieved from the UC Irvine Machine Learning Repository:

- Vehicle Silhouettes: eighteen quantitative conditional attributes, four decision classes, 846 objects – 592 training, 254 test set [17]. The goal of the data was to classify a given silhouette as one of four vehicle types, using a set of characteristics extracted from the silhouette. The vehicle can be viewed from one of many different angles. The images were acquired by a camera looking down at the vehicle model from a fixed elevation angle. The vehicles were rotated and their orientation angle was measured using a radial grid placed under the vehicle.
- Landsat Satellite: thirty-six quantitative conditional attributes, six decision classes, 6435 objects – 4435 training, 1000 test set [6]. Multispectral pixel values in a 3×3 neighborhood in a satellite image and a classification associated with the central pixel in each neighborhood. Each row of data corresponds to a neighborhood of pixels in a 3×3 square completely contained within an 82×100 sub-area. Each row contains the pixel values in the four spectral bands of each of the 9 pixels in the 3×3 neighborhood and a number indicating the classification label (earth type: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil) of the center pixel.
- Dry Bean: seventeen quantitative conditional attributes, seven decision classes, 13611 objects – 9527 training, 4084 test set [9]. The goal of the data was to classify type of beans based on the characteristics obtained from the image. Images of 7 different registered dry beans were taken with a high-resolution camera. Bean images obtained by computer vision systems were subjected to segmentation and feature extraction stages, and a total of 16 features – 12 dimensions and 4 shape forms were obtained from the grains.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

Each of the datasets were originally available as a single decision table. However, the proposed system explored the possibilities of classification based on fragmentary data. Therefore, the data was preprocessed – randomly dispersed. Different numbers of local tables were considered during dispersion. The data was divided into $3, 5, 7, 9$ and 11 local tables. Each table contained a reduced set of conditional attributes and all objects from the original table. Some attributes were common between local tables. It should also be noted that, the more local tables, the fewer attributes in each local table. The data was imbalanced – the number of objects in individual decision classes, both in the training and test sets, varied strongly. The specific cardinality are presented in Figure 1. Two variants for each of the datasets were considered in the study. Experiments were performed both on dispersed imbalanced data and on data that had been modified by applying one of the known methods for imbalanced data. The Synthetic Minority Over-sampling Technique (SMOTE) method was used in the paper [4]. This is an over-sampling method which adds artificially created objects to a dataset. The objects were created based on randomly selected minority class objects. For this purpose, the $k-$nearest neighbors algorithm was used. On the line connecting the selected object with its closest neighbors, a new object from the minority class was created. The implementation of this algorithm available in WEKA [11] software was used. Each local table was balanced separately. Each decision class except the most numerous one, was changed using SMOTE method in such a way that all decision classes had the same number of objects after balancing. Thus, in the end, we obtained 30 dispersed datasets: Vehicle Silhouettes,

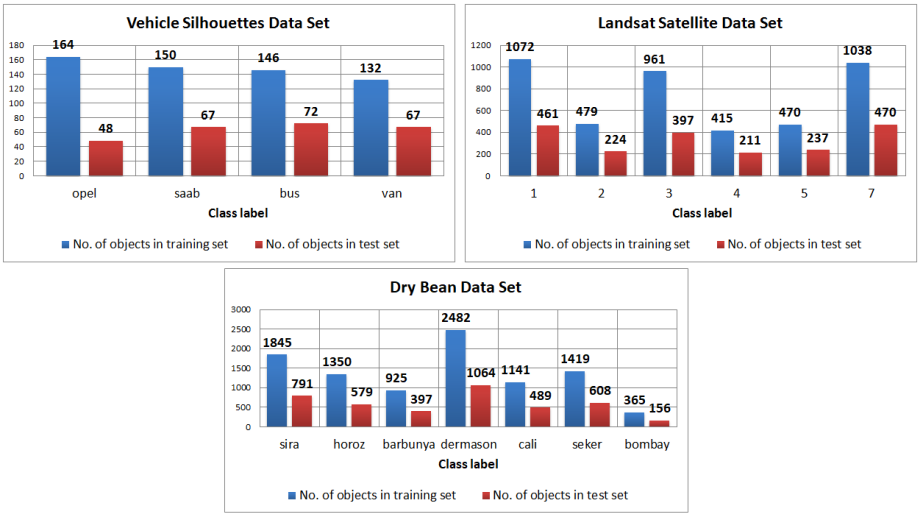

Fig. 1. Imbalance of data – cardinality of decision classes in training and test sets.

Vehicle Silhouettes balanced, Landsat Satellite, Landsat Satellite balanced, Dry

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

Bean, Dry Bean balanced; and for each dataset five versions of the dispersion: 3, 5, 7, 9, 11 local tables. The quality of classification was evaluated based on the test set. A classification accuracy measure (*acc*) was used for this purpose. That is, a fraction of the total number of objects in the test set that were classified correctly. As was mentioned before, a 10-fold cross-validation was used on the test set, i.e., the neural network was trained 10 times with 9 folds and tested on one remaining fold. In addition, each test was performed three times to ensure that the results were reliable and not distorted by the influence of randomness. The results for the neural network approach that are given below is the average of the obtained results.

Three approaches were tested and the results are presented below. The two tested approaches use the RBF networks described above. The first RBF approach used a centers training stage by means of back-propagating the error up to the centers during training whiles the second RBF approach used only the Lloyd's $k-$means algorithm to determine the centers. As a baseline approach, the approach proposed in [10] was used. This ensemble of classifiers method consists of creating three base classifiers: $k-$nearest neighbors, decision tree and naive bayes classifier (KNN, DT, NB) based on each local table. The approach was implemented in python programming language using implementations available in the sklearn library. Based on each local table, the three classifiers were built. The final decision was made using soft voting for all classifiers. Different parameter values were tested for approaches based on RBF networks. For the generation of prediction vectors using the $k-$nearest neighbors classifier, $k \in \{1, 5, 10\}$ parameters were studied. Due to the limited space of this paper, only the results for the optimal k value is presented. Different numbers of neurons in the hidden layer were tested for RBF neural networks. The following values $\{0.25, 0.5, 0.75, 1, 1.5, 1.75, 2, 2.5, 2.75, 3, 3.5, 3.75, 4, 4.5, 4.75, 5\}\times$ the number of neurons in the input layer were tested. The results obtained for the RBF network with a centers training stage is presented in Table 1. The results obtained for the RBF network with the Lloyd's algorithm is shown in Tables 2. The best result for each dispersed dataset (each line) is shown in bold. As mentioned early on, different values of the parameter k were analyzed $k \in \{1, 5, 10\}$. In the tables, only the results obtained for $k = 5$ are presented because this value was optimal – in most cases for this value the best results were obtained. The results obtained for the ensemble of classifiers approach is given in Table 3.

Let us begin the analysis of the proposed RBF network with a centers training stage approach and RBF network with the Lloyd's algorithm by comparing the complexity of the neural nets for which optimal results were generated. As can be seen, significantly lower network complexity is sufficient to achieve the best results for the proposed approach. Figure 2 gives a comparison of the minimum number of neurons in the hidden layer sufficient to achieve the optimal result. The reduction in network's complexity using the proposed approach is significant compared to using the Lloyd's algorithm.

Now, we compare the results of classification accuracy obtained using the three analyzed approaches. Table 3 summarizes all results – the best results ob-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

Table 1. Results of classification accuracy *acc* for the RBF network with a centers training stage. Designation I is used for the number of neurons in the input layer.

Dataset	No.	No. of neurons in hidden layer												
optimal k parameter	tables	0.25×I	0.5×I	0.75×I	1×I	1.5×I	1.75×I	2×I	2.5×I	2.75×I	3×I	3.5×I	3.75×I	4×I	4.5×I	4.75×I	5×I
Vehicle	3	0.483	0.608	0.642	0.661	0.702	0.719	0.725	0.735	0.734	0.759	0.756	0.763	0.752	**0.769**	0.762	0.765
imbalanced	5	0.568	0.651	0.667	0.683	0.716	0.724	0.733	0.75	0.754	0.758	0.754	0.743	0.75	0.749	0.749	**0.762**
k=5	7	0.567	0.639	0.671	0.668	0.685	0.702	**0.709**	0.7	0.701	0.694	0.685	0.682	0.69	0.68	0.682	0.684
	9	0.634	0.664	0.7	0.721	**0.73**	0.707	0.706	0.709	0.691	0.7	0.681	0.672	0.671	0.65	0.643	0.637
	11	0.6	0.654	0.69	0.707	0.711	0.702	0.707	0.705	**0.718**	0.703	0.714	0.714	0.707	0.712	0.707	0.702
Vehicle	3	0.721	0.735	0.741	0.735	0.73	0.744	0.752	0.752	0.748	0.759	0.756	0.744	0.76	0.755	0.755	**0.764**
balanced	5	0.709	0.72	0.755	0.757	0.754	**0.759**	0.749	0.745	0.755	0.745	0.745	0.734	0.73	0.725	0.719	0.724
k=5	7	0.726	0.727	**0.734**	0.714	0.722	0.706	0.707	0.7	0.684	0.684	0.667	0.658	0.653	0.629	0.625	0.62
	9	0.723	**0.732**	0.718	0.701	0.68	0.685	0.681	0.672	0.654	0.637	0.626	0.628	0.606	0.609	0.593	0.57
	11	**0.727**	**0.727**	0.72	0.722	0.702	0.698	0.693	0.668	0.668	0.659	0.647	0.631	0.597	0.593	0.58	0.588
Satellite	3	0.788	0.834	0.849	0.854	0.874	0.88	0.885	0.891	0.892	0.896	0.894	0.896	0.896	**0.897**	0.894	0.896
imbalanced	5	0.827	0.845	0.855	0.866	0.879	0.883	0.885	0.887	0.888	0.888	0.886	0.891	0.888	0.886	**0.892**	**0.892**
k=5	7	0.835	0.855	0.87	0.877	0.884	0.883	0.886	**0.887**	0.884	0.883	0.884	**0.887**	0.884	0.883	0.882	0.882
	9	0.845	0.857	0.87	0.877	0.884	**0.889**	0.883	0.886	0.883	0.885	0.883	0.88	0.882	0.882	0.881	0.882
	11	0.843	0.856	0.864	0.87	0.875	0.877	0.877	0.874	**0.878**	0.874	0.874	0.876	0.872	0.87	0.865	0.868
Satellite	3	0.659	0.775	0.813	0.832	0.852	0.856	0.858	0.864	0.864	0.867	0.867	0.868	0.872	0.873	**0.875**	0.874
balanced	5	0.753	0.808	0.839	0.849	0.85	0.855	0.854	0.858	0.858	0.863	0.862	0.864	**0.866**	0.865	**0.866**	**0.866**
k=5	7	0.78	0.834	0.852	0.851	0.85	0.856	0.859	0.861	**0.862**	0.861	0.861	0.86	0.861	0.86	0.859	0.859
	9	0.801	0.843	0.847	0.849	0.857	0.859	0.861	0.86	0.859	0.861	**0.863**	0.86	0.86	0.861	0.855	0.855
	11	0.815	0.843	0.846	0.845	0.851	0.851	0.855	0.859	**0.861**	0.859	0.857	0.854	0.85	0.849	0.848	0.846
Dry Bean	3	0.786	0.857	0.883	0.895	0.907	0.907	0.911	0.914	0.916	0.916	0.917	0.916	0.918	0.919	**0.92**	**0.92**
imbalanced	5	0.851	0.893	0.904	0.912	0.915	0.916	0.917	**0.918**	0.917	**0.918**	0.917	**0.918**	0.917	0.917	0.917	0.917
k=5	7	0.871	0.903	0.913	0.914	0.916	0.916	**0.917**	**0.917**	0.916	**0.917**	**0.917**	**0.917**	**0.917**	0.916	0.916	0.915
	9	0.881	0.905	0.912	0.915	0.917	0.917	0.917	0.918	**0.919**	0.918	0.918	0.918	0.917	0.918	0.918	0.918
	11	0.888	0.908	0.912	0.915	**0.916**	0.914	**0.916**	0.915	**0.916**	0.915	0.915	**0.916**	**0.916**	0.914	0.914	0.915
Dry Bean	3	0.789	0.855	0.882	0.892	0.902	0.908	0.91	0.913	0.916	0.916	**0.918**	**0.918**	**0.918**	**0.918**	0.917	**0.918**
balanced	5	0.852	0.892	0.905	0.91	0.915	0.916	**0.918**	**0.918**	0.917	**0.918**	0.917	0.916	0.917	0.916	0.916	0.917
k=5	7	0.865	0.901	0.91	0.914	**0.917**	**0.917**	0.915	0.916	0.916	**0.917**	0.916	0.916	**0.917**	**0.917**	0.914	0.915
	9	0.882	0.906	0.911	0.916	0.918	**0.919**	0.918	0.918	**0.919**	0.917	0.918	0.917	0.918	0.915	0.917	0.917
	11	0.886	0.906	0.911	0.915	0.917	**0.918**	0.916	0.916	0.917	0.916	0.917	0.917	0.916	0.914	0.915	0.916

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

Table 2. Results of classification accuracy *acc* for the RBF network with the Lloyd's algorithm. Designation I is used for the number of neurons in the input layer.

Data set	No.	No. of neurons in hidden layer												
optimal k parameter	tables	0.25×I	0.5×I	0.75×I	1×I	1.5×I	1.75×I	2×I	2.5×I	2.75×I	3×I	3.5×I	3.75×I	4×I	4.5×I	4.75×I	5×I
Vehicle	3	0.268	0.351	0.427	0.476	0.54	0.55	0.571	0.588	0.613	0.629	0.631	0.639	0.642	0.641	0.65	**0.656**
imbalanced	5	0.289	0.429	0.487	0.529	0.57	0.574	0.612	0.625	0.638	0.631	0.657	0.657	0.671	0.67	0.688	**0.692**
k=5	7	0.341	0.424	0.479	0.523	0.566	0.597	0.603	0.633	0.638	0.638	0.656	0.661	0.669	0.671	0.672	**0.676**
	9	0.363	0.496	0.545	0.582	0.616	0.623	0.637	0.659	0.655	0.665	0.686	0.688	0.716	0.719	0.719	**0.727**
	11	0.359	0.469	0.537	0.575	0.601	0.61	0.623	0.651	0.647	0.659	0.674	0.683	0.687	0.687	0.686	**0.698**
Vehicle	3	0.556	0.692	0.689	0.691	0.722	0.722	0.742	0.738	0.746	0.731	0.741	0.746	0.75	0.751	0.75	**0.761**
balanced	5	0.509	0.618	0.643	0.659	0.692	0.719	0.71	0.718	0.731	0.735	0.744	**0.757**	0.751	0.753	0.748	0.747
k=5	7	0.683	0.695	0.702	0.71	0.735	0.717	0.739	0.731	0.733	0.725	0.731	**0.741**	0.739	0.734	0.738	0.738
	9	0.503	0.671	0.674	0.687	0.707	0.709	0.717	0.72	0.721	**0.729**	0.728	0.72	0.724	0.725	0.725	0.72
	11	0.561	0.611	0.637	0.691	0.71	0.715	0.713	**0.722**	**0.722**	0.715	0.705	0.709	0.701	0.701	0.694	0.694
Satellite	3	0.729	0.789	0.817	0.823	0.83	0.833	0.834	0.841	0.846	0.846	0.85	0.848	0.851	0.851	0.852	**0.853**
imbalanced	5	0.759	0.809	0.828	0.83	0.838	0.84	0.845	0.849	0.85	0.852	0.852	0.852	0.855	0.855	**0.859**	**0.859**
k=5	7	0.773	0.825	0.834	0.833	0.841	0.846	0.848	0.851	0.853	0.853	0.854	0.858	0.858	0.859	0.859	**0.863**
	9	0.814	0.832	0.832	0.839	0.841	0.845	0.843	0.849	0.849	0.851	0.857	0.857	0.859	0.861	0.861	**0.862**
	11	0.81	0.831	0.841	0.841	0.844	0.847	0.848	0.85	0.851	0.851	0.854	0.855	0.856	0.857	**0.86**	0.859
Satellite	3	0.56	0.674	0.72	0.757	0.791	0.804	0.81	0.818	0.825	0.829	0.837	0.84	0.843	0.846	0.848	**0.849**
balanced	5	0.629	0.714	0.758	0.787	0.815	0.819	0.822	0.839	0.841	0.842	0.843	0.845	0.848	**0.85**	0.847	0.847
k=5	7	0.644	0.75	0.799	0.809	0.823	0.829	0.838	0.843	0.843	0.843	0.845	0.846	0.849	0.85	**0.851**	0.85
	9	0.669	0.774	0.8	0.818	0.828	0.832	0.843	0.843	0.844	0.841	0.846	**0.847**	0.846	0.843	0.844	0.845
	11	0.714	0.789	0.814	0.82	0.834	0.837	0.837	0.841	0.844	0.846	0.849	0.851	0.848	**0.851**	0.85	0.849
Dry Bean	3	0.591	0.777	0.828	0.842	0.864	0.869	0.871	0.882	0.886	0.887	0.891	0.892	0.895	0.897	**0.9**	0.898
imbalanced	5	0.762	0.833	0.858	0.874	0.89	0.893	0.894	0.897	0.899	0.9	0.902	0.904	0.905	0.905	0.904	**0.906**
k=5	7	0.796	0.847	0.873	0.883	0.893	0.896	0.9	0.901	0.904	0.903	0.905	0.905	0.906	0.907	0.908	**0.909**
	9	0.817	0.856	0.877	0.885	0.892	0.894	0.897	0.899	0.9	0.901	0.901	0.901	0.904	0.905	**0.906**	**0.906**
	11	0.827	0.867	0.886	0.891	0.898	0.9	0.9	0.901	0.901	0.902	0.901	0.902	0.902	0.903	0.904	**0.905**
Dry Bean	3	0.58	0.774	0.818	0.841	0.859	0.873	0.876	0.884	0.885	0.889	0.891	0.892	0.893	0.893	**0.896**	0.894
balanced	5	0.766	0.835	0.854	0.87	0.886	0.891	0.892	0.895	0.898	0.9	0.899	0.903	0.902	0.901	0.903	**0.905**
k=5	7	0.798	0.852	0.869	0.881	0.891	0.894	0.898	0.9	0.901	0.901	0.902	0.905	0.906	0.907	0.908	**0.91**
	9	0.817	0.859	0.873	0.885	0.894	0.896	0.897	0.9	0.9	0.901	0.902	0.903	0.905	0.906	**0.907**	0.905
	11	0.828	0.868	0.882	0.889	0.896	0.897	0.9	0.901	0.902	0.902	0.901	0.902	0.903	**0.904**	**0.904**	**0.904**

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

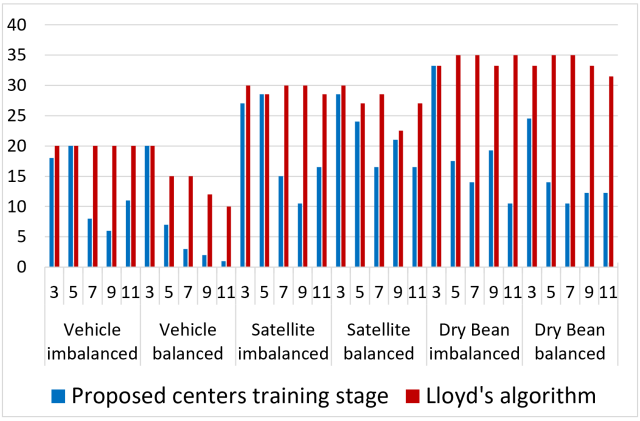

Fig. 2. The minimum number of neurons in the hidden layer sufficient to achieve the optimal result – comparison of RBF networks with a centers training stage approach and with the Lloyd's algorithm.

tained for the proposed approach, the best results using the RBF networks with the Lloyd's algorithm and the results for the ensemble of classifiers approach. The best result for each dispersed dataset (each line) is shown in bold. As can be seen, in all cases (except one) the best results were generated using the proposed approach. Statistical tests were performed in order to confirm significant differences in the obtained results *acc*. The received classification accuracy were divided into three dependent data samples, results from Table 3. The Friedman test was used to detect differences in multiple test samples. There was a statistically significant difference in the results obtained for the three different approaches being considered, $\chi^2(29, 2) = 39.467, p = 0.000001$. Additionally, comparative box-whiskers charts for the results with three approaches were created (Fig. 3). As can be observed, the values of the classification accuracy for the proposed approach – RBF network with a centers training stage is the best (much better than the others approaches). In the next step, the Wilcoxon each-pair test was used. This test confirmed that the differences in the classification accuracy were significant between the RBF network with a centers training stage and both the RBF network with the Lloyd's algorithm and the ensemble of classifiers approach. There in no statistically significant difference in the classification accuracy between the RBF network with the Lloyd's algorithm and the ensemble of classifiers approach.

For the proposed approach, a comparison of the results obtained by using balanced and imbalanced datasets were also made. Figure 4 shows the comparison of the results in a bar chart and box-whiskers charts. As can be seen for the Dry Bean set, balancing the dataset had no effect on the results. For the Satellite set, better results were obtained for the imbalanced dataset. On the other hand, for the Vehicle dataset, for a smaller number of local tables (3 and 5 tables) we got

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

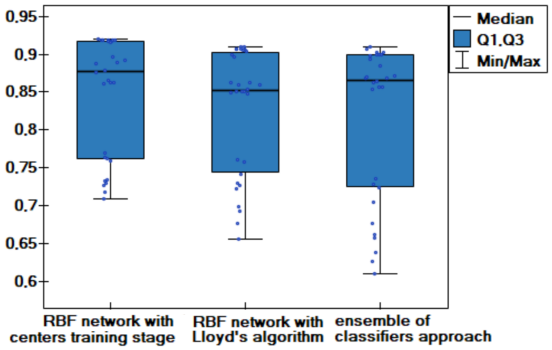

Fig. 3. Comparison of the results obtained for the three approaches: the RBF network with the centers training stage, the RBF network with the Lloyd's algorithm and the ensemble of classifiers approach.

better results for the imbalanced data. Also, the box-whiskers charts confirmed that there was no difference between the results for balanced and imbalanced datasets. The Wilcoxon test confirmed that there was no statistically significant difference in the average classification accuracy obtained for balanced and imbalanced sets for the proposed approach. Thus, it can be concluded that the proposed approach performs very well for imbalanced data.

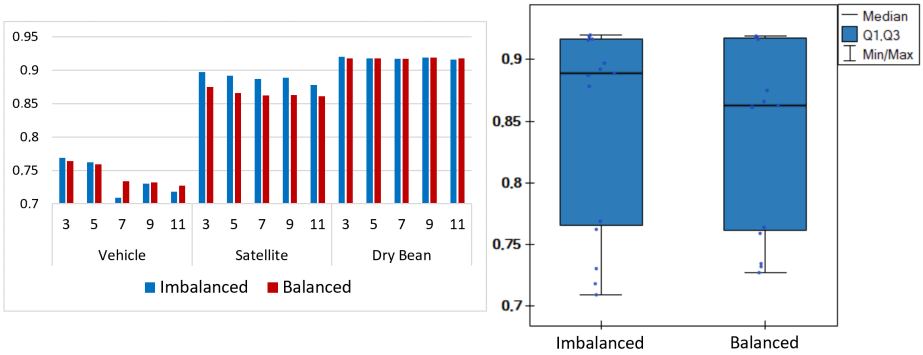

Fig. 4. Comparison of the results obtained for the RBF network with a centers training stage and imbalanced versus balanced datasets.

For the proposed approach, a comparison of the results obtained for different versions of dispersion was made. Figure 5 shows the comparison of the results in a bar chart and box-whiskers charts. As can be seen for the Dry Bean dataset, degree of dispersion of the dataset had no effect on the results. For the Satellite and the Vehicle datasets, better results were obtained for a smaller number of

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

Table 3. Comparison of classification accuracy *acc* obtained for approaches: the RBF network with a centers training stage, the RBF network with the Lloyd's algorithm and the ensemble of classifiers approach from paper [10].

Data set	No. tables	RBF network with a centers training stage	RBF network with Lloyd's algorithm	ensemble of classifiers from paper [10]
Vehicle imbalanced k=5	3	**0.769**	0.656	0.657
	5	**0.762**	0.692	0.638
	7	**0.709**	0.676	0.626
	9	**0.73**	0.727	0.661
	11	**0.718**	0.698	0.61
Vehicle balanced k=5	3	**0.764**	0.761	0.728
	5	**0.759**	0.757	0.724
	7	0.734	**0.741**	0.736
	9	**0.732**	0.729	0.705
	11	**0.727**	0.722	0.677
Satellite imbalanced k=5	3	**0.897**	0.853	0.885
	5	**0.892**	0.859	0.872
	7	**0.887**	0.863	0.868
	9	**0.889**	0.862	0.868
	11	**0.878**	0.86	0.863
Satellite balanced k=5	3	**0.875**	0.849	0.87
	5	**0.866**	0.85	0.864
	7	**0.862**	0.851	0.856
	9	**0.863**	0.847	0.856
	11	**0.861**	0.851	0.854
Dry Bean imbalanced k=5	3	**0.92**	0.9	0.906
	5	**0.918**	0.906	0.902
	7	**0.917**	0.909	0.899
	9	**0.919**	0.906	0.894
	11	**0.916**	0.905	0.9
Dry Bean balanced k=5	3	**0.918**	0.896	0.909
	5	**0.918**	0.905	0.899
	7	**0.917**	0.91	0.9
	9	**0.919**	0.907	0.898
	11	**0.918**	0.904	0.903

local tables – a smaller degree of dispersion, but the differences were not large. The Wilcoxon test confirmed that there was statistically significant differences in the average classification accuracy only between pairs of dispersion: 3 and 7 local tables; 5 and 7 local tables; 9 and 11 local tables. The conclusion of this comparison is that the proposed approach handles both small and large data dispersion quite well which is a very important property because often in real situations we have to deal with large dispersion – many units providing independent datasets. The proposed method requires optimization of several parameters in both the k-nearest neighbors classifier and the neural network, which can be considered a drawback of the method.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

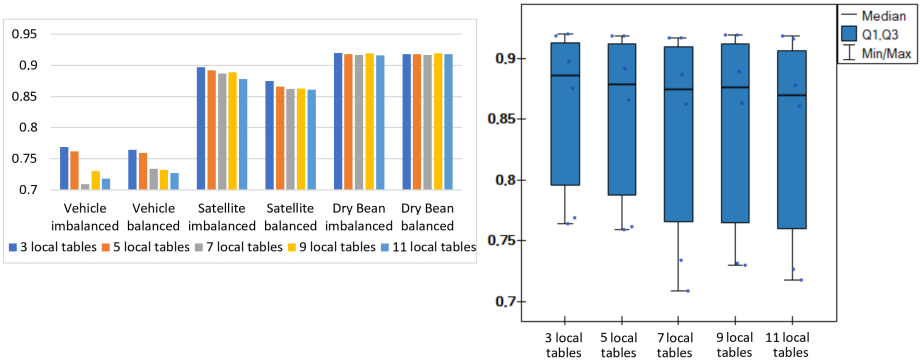

Fig. 5. Comparison of the results obtained for the RBF network with a centers training stage and different dispersion versions of datasets – 3, 5, 7, 9 and 11 local tables.

4 Conclusion

The study concerned analysis of the classification problem of an object presented in an image based on the characteristics of the object stored in a fragmentary form – a set of local decision tables. For this purpose, a RBF network model with a centers training stage was proposed. The paper shows that this approach gives much better results than the RBF network model with the Lloyd's algorithm. In addition, the network's structure is much simpler for the proposed approach. Moreover, the proposed approach gives better results than the ensemble of classifiers approach. It was also shown that the proposed model copes very well with imbalanced datasets and that the degree of dispersion does not have a large impact on classification accuracy. In future works, it is planned to use neural networks to define predictions based on local tables. The possibility of using a global learning stage after building the RBF network that combines predictions is also being considered. This stage would be implemented by using some artificially generated data.

References

1. Basha, S. S., Dubey, S. R., Pulabaigari, V., Mukherjee, S.: Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing, 378, 112–119, (2020)
2. Caggiano, A., Zhang, J., Alfieri, V., Caiazzo, F., Gao, R., Teti, R.: Machine learning-based image processing for on-line defect recognition in additive manufacturing. CIRP annals, 68(1), 451–454, (2019)
3. Chaki, J., Dey, N.: Fragmented handwritten digit recognition using grading scheme and fuzzy rules. Sādhanā, 45(1), 1–23, (2020)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res., 16:321–357, 2002.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7

5. Chen, J., Liu, G., Chen, X.: AnimeGAN: a novel lightweight GAN for photo animation. In International symposium on intelligence computation and applications 242–256. Springer, Singapore, (2020)
6. Dua, D. and Graff, C.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, (2019)
7. Fornasier, M., Toniolo, D.: Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images. Pattern Recognition, 38(11), 2074–2087, (2005)
8. Giuseppe, A., Falchi, F.: k-NN based image classification relying on local feature similarity. International Conference on SImilarity Search and Applications, SISAP '10, Istanbul, Turkey.
9. Koklu, M., Ozkan, I. A.: Multiclass classification of dry beans using computer vision and machine learning techniques. Computers and Electronics in Agriculture, 174, 105507, (2020)
10. Kurian, R. A., Lakshmi, K.: An ensemble classifier for the prediction of heart disease. International Journal of Scientific Research in Computer Science, 3 , 25–31, (2018)
11. Markov, Z., Russell, I.: An introduction to the WEKA data mining system. SIGCSE Bull. 38, 3 (2006), 367–368. https://doi.org/10.1145/1140123.1140127
12. Mozgovoy, D. K., Hnatushenko, V. V., Vasyliev, V. V.: Automated recognition of vegetation and water bodies on the territory of megacities in satellite images of visible and IR bands. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4(3), 167-172, (2018)
13. Przybyła-Kasperek, M., Wakulicz-Deja, A.: Global decision-making system with dynamically generated clusters, Inform. Sci. 270, 172–191 (2014)
14. Ptucha, R., Such, F. P., Pillai, S., Brockler, F., Singh, V., Hutkowski, P.: Intelligent character recognition using fully convolutional neural networks. Pattern recognition, 88, 604–613, (2019)
15. Santos, I., Castro, L., Rodriguez-Fernandez, N., Torrente-Patino, A., Carballal, A.: Artificial neural networks and deep learning in the visual arts: A review. Neural Computing and Applications, 33(1), 121–157, (2021)
16. Shelepin, Y. E., Chikhman, V. N., Foreman, N.: Analysis of the studies of the perception of fragmented images: global description and perception using local features. Neuroscience and behavioral physiology, 39(6), 569–580, (2009)
17. Siebert, J. P.: Vehicle Recognition Using Rule Based Methods, Turing Institute Research Memorandum TIRM-87-0.18, March 1987.
18. Sochor, J., Špaňhel, J., Herout, A.: Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. IEEE Transactions on Intelligent Transportation Systems, 20(1), 97–108, (2018)
19. Traore, B. B., Kamsu-Foguem, B., Tangara, F.: Deep convolution neural network for image recognition. Ecological Informatics, 48, 257–268, (2018)
20. Wan, S., Goudos, S.: Faster R-CNN for multi-class fruit detection using a robotic vision system. Computer Networks, 168, 107036, (2020)
21. Yadav, S. S., Jadhav, S. M.: Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data, 6(1), 1–18, (2019)
22. Yang, R., Sarkar, S.: Gesture recognition using hidden markov models from fragmented observations. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) vol. 1, 766–773. IEEE, (2006)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_7