

# Classification methods for handwritten digit recognition: A survey

Ira M. Tuba<sup>a</sup>, Una M. Tuba<sup>b</sup>, Mladen Đ. Veinović<sup>b</sup>

Singidunum University, Faculty of Informatics and Computing,  
Belgrade, Republic of Serbia

<sup>a</sup>e-mail: ituba@ieee.org,

ORCID iD: <https://orcid.org/0000-0002-2863-7084>

<sup>b</sup>e-mail: una.svetlana.tuba.17@singimail.rs,

ORCID iD: <https://orcid.org/0000-0002-4224-9248>

<sup>c</sup>e-mail: mveinovic@singidunum.ac.rs, **corresponding author**,

ORCID iD: <https://orcid.org/0000-0001-6136-1895>

DOI: 10.5937/vojtehg71-36914; <https://doi.org/10.5937/vojtehg71-36914>

FIELD: mathematics, computer sciences

ARTICLE TYPE: review paper

## Abstract:

*Introduction/purpose: This paper provides a survey of handwritten digit recognition methods tested on the MNIST dataset.*

*Methods: The paper analyzes, synthesizes and compares the development of different classifiers applied to the handwritten digit recognition problem, from linear classifiers to convolutional neural networks.*

*Results: Handwritten digit recognition classification accuracy tested on the MNIST dataset while using training and testing sets is now higher than 99.5% and the most successful method is a convolutional neural network.*

*Conclusions: Handwritten digit recognition is a problem with numerous real-life applications. Accurate recognition of various handwriting styles, specifically digits is a task studied for decades and this paper summarizes the achieved results. The best results have been achieved with convolutional neural networks while the worst methods are linear classifiers. The convolutional neural networks give better results if the dataset is expended with data augmentation.*

*Key words: handwritten digit recognition, image classification, support vector machine, deep neural networks, convolutional neural networks, hyperparameter optimization, swarm intelligence, MNIST.*

## Introduction

Nowadays, computers can be considered an essential part of our lives; they changed the way of handling problems in everyday life as well as in sci-



ence. They are capable of processing information much faster compared to humans. However, the ways of processing data in the human brain and in the computer are completely different. While computers can process billions of mathematical operations in a second and are incomparably better than humans in that area, there are tasks where computers are significantly inferior. Recognizing different objects such as identification of people, for example, is a task that humans do in a fraction of a second. Regardless of the variety of lighting conditions, angles of observation, hairstyles, etc. identifying people is not challenging for humans and it does not require any special education, capability, or skills for that. On the other hand, explaining to the computer how to identify people is a really difficult task.

The complexity of the object recognition problem can be illustrated by the case when only a limited number of objects, e.g. digits and letters, should be recognized by a computer. This problem has practical applications such as license plate recognition. One of the most intuitive approaches for this recognition would be template matching. Template matching is a technique that requires a database of images of objects, in this case, digits and letters. Recognition is done by comparing an image of an unknown object with the images from the database and looking for the closest match. It is a simple task, a simple image comparison. However, the problem with this simple solution is that when it comes to practical application, images of digits and letters could be taken under different angles, light can be different, license plates can be covered by dirt, etc.; hence a database should contain a huge number of such images and matching becomes very difficult and time consuming. Due to the mentioned reasons, it gives very poor results in practice.

To get better results, it is necessary to present data differently. A raw image is not very informative to a computer, and it would be more efficient if numerical data that represent certain characteristics or features of those images were used. For example, a histogram of projections is one simple feature that could be used to recognize regularly shaped characters (letters and digits) (Fig. 1 presented in (Nosseir & Roshdy, 2018)). The number of required computations is significant and humans certainly would not use that method, but that suits the computer and enables it to recognize characters much more precisely. Other features such as projections under different angles, contours, edges, etc. can be added to increase recognition accuracy. Computer recognition of regularly shaped characters is a

challenging task but appropriate methods were proposed a long time ago and nowadays there are numerous practical applications where they are used (e.g. license plate recognition at toll gates, traffic control, etc.).

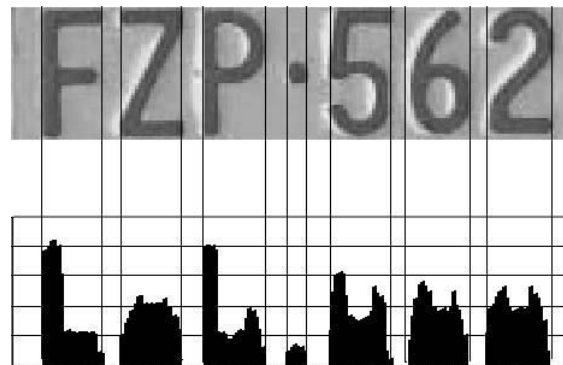


Figure 1 – Histogram of the projections on the  $x$ - and  $y$ -axes  
(Nosseir & Roshdy, 2018)

Рис. 1 – Гистограмма проекций по осям  $x$  и  $y$  (Nosseir & Roshdy, 2018)

Слика 1 – Хистограм пројекција на  $x$ - и  $y$ - осу (Nosseir & Roshdy, 2018)

The next level of difficulty would be the recognition of handwritten characters. Even though this sounds like a similar problem to the previous one, it is significantly more complex. Reading handwriting is often challenging even for a human due to a large number of different handwriting styles that are sometimes rather illegible. Since this is a very complicated problem, the recognition of only handwritten digits, without letters, was studied as a separate problem. The need for applications that recognize handwritten digits appears for automatic sorting of mail with zip codes or for sorting bank checks. Fig. 2 presents a variety of writing styles where it can be seen that some handwritten digits are difficult to identify.

Handwritten digit recognition is an old practical problem that has been researched for decades. Initial results were not acceptable for practical use while nowadays even real time methods for handwritten character recognition are widely available and successfully used for numerous purposes.

This paper reviews the methods for handwritten digit recognition. First, the problem formulation and the benchmark dataset for handwritten digit recognition are presented. Next, the classification methods and the proposed features from literature are described. After that, the classifiers and the features mentioned in this paper along with the obtained classification

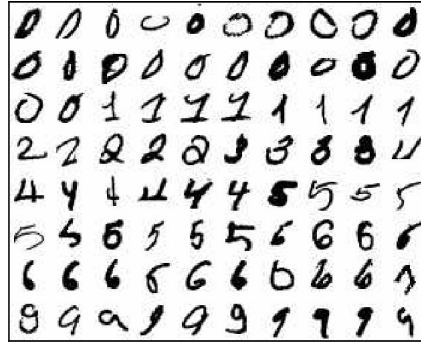


Figure 2 – Examples of digits hard to recognize  
 Рис. 2 – Примеры труднораспознаваемых цифр  
 Слика 2 – Примери цифара тешких за препознавање

error rates are described and summarized in a table. Finally, the conclusions about the quality of different classifiers and features are presented.

### Handwritten digit recognition - a classification problem

Handwritten digit recognition is a classification problem which is a supervised machine learning task. Machine learning is a subset of artificial intelligence algorithms used for instructing computers how to learn from given data instead of relying on explicit specification of steps. The main tasks of machine learning are unsupervised, supervised and reinforcement learning. In unsupervised learning, the computer uses the given data and searches for a pattern in them. The result should be the data grouped based on their similarities. On the other hand, in supervised learning, the groups or classes are known, given by the user, and the task is to create a model that will be able to label each instance with the correct class. In order to do that, a training set, and set of instances with their corresponding labels are needed so the algorithm can learn the pattern. Reinforcement learning methods are based on rewards for desirable actions and penalties for undesirable ones. It teaches an intelligent agent which action to take in order to maximize the reward.

The classification problem has been studied widely for several decades. This resulted in numerous different classification algorithms where each of them has advantages over others for certain types of problems. The simplest classifiers include linear classifiers where the decision is made

based on the linear combination of inputs. Linear classifiers are good for problems where instances are linearly separable which is rarely a case for real life problems. Non-linear classifiers are more applicable and nowadays there are many of them. Some of widely used non-linear classifiers are k-nearest neighbors, support vector machines, artificial neural networks, convolutional neural networks, etc.

Handwritten digit recognition has been widely studied for decades and with the development of classification methods, the accuracy of handwritten digit recognition has increased. For digital image classification problems such as handwritten digit recognition, usually, instead of a raw image as an input, various image features were extracted and used.

## MNIST dataset

Considering the long history of studying this problem, handwritten digit recognition has become a benchmark problem for testing new machine learning classifiers. In order to enable a comparison between different methods for handwritten digit recognition, the MNIST dataset was proposed. The MNIST (Modified National Institute of Standards and Technology) is a standard benchmark dataset of handwritten digit images made in 1998 (Lecun et al., 1998). It was made by regrouping, selecting, and resizing images from two different special NIST datasets. Digits in the first special NIST dataset were written by high school students and the second dataset contains digits written by the employees of the United States Census Bureau. Usually, for the training set, only the digits written by the employees were used and the model was tested on the digits written by high school students. The images in the MNIST dataset were mixed from both sets and the test set does not contain the digits written by the same person as in the training set. The originally binary images from the NIST were normalized to size 20x20 while preserving the aspect ratio. Due to anti-aliasing during the normalization process, the resulting images are gray scale images instead of binary ones. The final images were at the end boxed so the size is 28x28 pixels.

The MNIST database consists of 70,000 images in total, divided into training and test sets. The training set consists of 60,000 images while the testing set has the remaining 10,000 images of handwritten digits. The distribution of the number of images of each digit in the training and test sets is presented in Fig. 3.

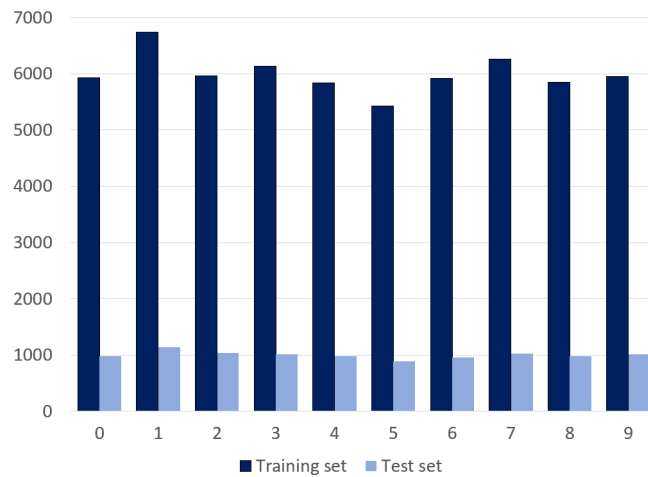


Figure 3 – Distribution of the digits in the MNIST dataset  
 Рис. 3 – Распределение цифр в наборе данных MNIST  
 Слика 3 – Дистрибуција цифара у МНИСТ скупу података

The examples of the images in the MNIST dataset are presented in Fig. 4.



Figure 4 – Examples of the images in the MNIST dataset  
 Рис. 4 – Примеры изображений в наборе данных MNIST  
 Слика 4 – Примери слика из МНИСТ скупа података

### Classification tested on the MNIST dataset

The results reported on the widely known and used *Kaggle* platform for data science and machine learning can give an idea of what has been researched so far about the handwritten digit recognition problem. For official purposes, only the results published in the recognized scientific journals

should be considered, but these results could be a good starting guidance for research.

Based on the data from 2018, the classification accuracy on the MNIST dataset was up to 99.8% while a higher accuracy is possible only if all data are used for creating a model and the reported results show the accuracy of the training process. The results are presented in Fig. 5 (Deotte, 2018).

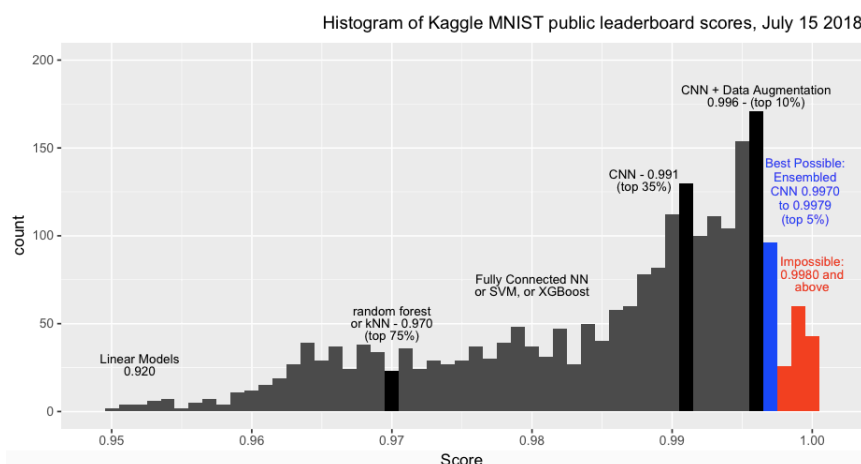


Figure 5 – Classification accuracy on the MNIST dataset (Deotte, 2018)

Рис. 5 – Точность классификации в наборе данных MNIST (Deotte, 2018)

Слика 5 – Тачност класификације на МНИСТ бази података (Deotte, 2018)

The graph presented in Fig. 5 shows the classification accuracy on the x-axis and the number of the reported results on the y-axis. One of the conclusions from the graph in Fig. 5 is that linear classifiers produced very poor accuracy and consequently were used less. Non-linear classifiers such as random forest and k-nearest neighbors are the second worse methods. The average classification accuracy was 97%. Artificial neural networks, support vector machines and XGBoost are slightly better for the handwritten digit recognition problem tested on the MNIST dataset. The average classification accuracy for these methods was 98%. A significant rise in the number of reported results is recorded for convolutional neural networks. The accuracy achieved by the CNN is higher than 99% and rises if the dataset is extended by data augmentation. In that case, the classification accuracy reaches up to 99.8%. The results that report higher classification accuracy were trained on the whole dataset which means that a training accuracy was reported, hence the comparison is not fair.

## Linear classifiers

Linear classifiers are machine learning algorithms that classify instances based on the linear combination of their features. Some examples of the linear classifiers are linear discriminant analysis, regularized discriminant analysis, naive Bayes, logistic regression, support vector machine (SVM) without kernel trick, artificial neural network (ANN) without hidden layers, etc. The advantage of these classifiers is significantly less time needed for training and testing compared to non-linear models. On the other hand, the classification accuracy for complex problems is usually smaller. For certain types of classification problems with many features and data that are linearly separable, such as document classification, linear classifiers showed great results (Huang & Lin, 2016; Yuan et al., 2012).

In general, linear classifiers are not suitable for handwritten digit recognition. The results obtained by them are significantly lower compared to non-linear methods which is the reason why there are not a lot of recent results reported. In (Lecun et al., 1998), the proposed MNIST benchmark dataset was trained by various machine learning methods. The error rate of the linear classifier was 12.00%. A set of linear classifiers that classify digits pairwise, nowadays known as the *one against one* method of multiclassification with binary classifiers, led to the improvement of the classification of handwritten digits so the error rate was 7.60%.

The results of linear classifiers were improved in (Ebrahimzadeh & Jampour, 2014) where a linear SVM was proposed. A histogram of oriented gradients was used as the feature vector. The error rate obtained by the proposed feature set and a linear SVM was 2.75% which according to the extensive search of the literature is the lowest error rate obtained with the linear classifier.

## Non-linear classifiers

The k-nearest neighbors (kNN) is a machine learning algorithm that can be used for classification. The kNN does not require any training and classification is done by a simple comparison with the available training data. The class of unknown instances is determined by the k nearest known instances, i.e. neighbors, where k is a positive, usually odd, integer. The class of the unknown instance is the same as the class of the majority of its neighbors. When using the kNN for classification, it is important to choose



the right distance metric and value for  $k$ . Smaller values for  $k$  reduce the reliability of the classification results while larger values for the parameter  $k$  can improve the classification accuracy but improvements are made up to a certain point and if the set value is too large, the error rate would start increasing again.

The major advantage of the kNN is its simplicity and the time saved for training. This can also be considered a disadvantage in some cases. Other classifiers would be trained just once and the generated model is later used for determining a class of the new instance. After a classification model is found, the classification of unknown instances is rather fast - it is only necessary to calculate the value of a more or less complex mathematical function. With the kNN, each new instance has to be compared with all instances in the training set which, depending on the dataset size, makes it not usable for real-time classification. In general, a larger training set is desirable for increasing the accuracy of the model and a better identification of outliers. Another component that affects the classification time is the number of features. However, classification by the kNN enables some kind of reinforcement learning since the training test grows with each new instance that is tested.

The kNN has been used for handwritten digit recognition with different feature sets. In (Grover & Toghi, 2020), padding of size 2 was added to the original images in the MNIST dataset so the images were of size 30x30. By using the sliding window method, eight new images were generated from each image in the training set. Instead of comparing a new instance with just original images from the training set, it was also compared with the newly generated images. This was done to reduce mistakes in classification due to small spatial translation in images. The Euclidean distance was used as a distance metric. The error rate of classification with the proposed method was 2.27%.

The kNN classification based on structural and statistical features of handwritten digit images was proposed in (Babu et al., 2014). The features included in the (Babu et al., 2014) are the number of holes in a digit image, area, height, width and width to height ratio of the digit in the images, horizontal and vertical crossing, branch, and cross points. The preprocessing step included binarization and noise removal. Based on 18 extracted features and the Euclidean distance and 5,000 test images, the error rate

obtained by the kNN classifier was 1.58%. The parameter  $k$  of the kNN was set to be 1.

Another approach that proposes kNN as a classifier was presented in (Ilmi et al., 2016). The feature set was based on well-known texture features, i.e. the local binary pattern (LBP). The LBP is chosen since it is invariant to monotonous variance on grayscale images. Additionally, a rotation invariant variant of the LPB was used. The proposed method was tested with the kNN where the  $k$  was 10. The minimal error rate was when all pixels of the digit image were used and it was 10.19%.

The naive Bayesian classifier has been used for pattern recognition and other classification problems for decades (Bhagya Shree & Sheshadri, 2018; Lyu et al., 2018; Zhai et al., 2015). It represents a machine learning method that assumes conditional independence between features. This assumption can decrease the probability of overfitting the model, but the issue that occurs for higher dimensional problems, i.e. classification of instances with numerous features, is numerical underflows. In (Wang & Zhang, 2020), usage of geometric means on likelihoods was proposed for solving this problem. The features of the instances given to the naive Bayesian classifier were the binarized pixel values of the images from the MNIST dataset. The authors in (Wang & Zhang, 2020) acknowledge the problem that the assumption needed for this classifier does not hold, but in order to test the proposed method of solving the numerical underflows problem, they used the MNIST dataset. The error rate was 18.60% which is very low for this dataset, but the classifier was applied on raw data instead of on the features. Better results were achieved with the naive Bayesian classifier when different features were used for training in (Armstrong, 2019). The feature set included the ark length which is the longest continuous set of pixels of the digit, the enclosed area or the number of pixels inside a closed contour, the number of contours, the histogram of oriented gradients, and the image moments. With this set of features, the naive Bayesian classifier achieved the classification error of 10.03%. In general, there are not a lot of papers that use this classifier since it cannot achieve results comparable to other machine learning algorithms.

## Support vector machine

The support vector machine (SVM) is a very powerful classifier that has been used in various applications for decades. The SVM has been applied

to classification problems in numerous areas such as medicine (Kang et al., 2019), (Wang et al., 2018), economy (Sivaram et al., 2020), signal processing (Esmaili et al., 2016), agriculture (Sethy et al., 2020), etc. Originally, the SVM was just a linear binary classifier where instances were presented as points in the space and based on them, the optimal hyperplane that separates points from different classes was found. The original idea of the SVM could not be used for solving most of real life problems since data had to be linearly separable. In order to make the SVM usable, it was adjusted to allow minor mistakes in the classification of the instances in the training set. Additionally, a kernel trick was introduced which made classification of non-linear data more efficient. This version of the SVM, with the slack variable for allowing misclassification and the kernel trick, was introduced in 1995 by Vapnik (Cortes & Vapnik, 1995) and it is still used.

The non-linear support vector machine was applied to the problem of handwritten digit recognition and achieved among the best results until the convolutional neural networks. One of the main issues when using the SVM is to find a good set of features that describes images. It is necessary to capture the similarity of images of the same digit and to distinguish different digits as easily as possible. Another problem when using the SVM is finding the optimal parameters for a kernel function.

Numerous different features were proposed and combined for digits in the MNSIT dataset. In (Patel & Kalyani, 2016), an inverse fringe distance map (IFDM) was proposed and the results were compared with the simple fringe distance map (FDM). The fringe distance map represents the distance of each pixel in the image from the closest black pixel. Before calculating the map, it is necessary to transform an image into a binary image, where the pixels of the digit are black and the background consists of white pixels. Finding the IFDM is the opposite - calculating the distance of each pixel from the nearest white pixel. Since images are 28x28, in total, the dimension of the feature vector is 784. The classification error was 2.86% when using the FDM, 2.28% for the IFDM. By combining the FDM and the IFDM into one feature vector, the error rate was 2.45%.

The support vector machine optimized by the swarm intelligence algorithm was proposed for the classification of the MNIST dataset where each digit was presented with the histograms of projections to four different axes:  $x = 0$ ,  $y = 0$ ,  $y = x$  and  $y = -x$  in (Tuba et al., 2016). The dimension of the feature vector was 166. The error rate was 4.40%.

In ([Gattal et al., 2014](#)), a support vector machine with the radial basis function kernel was used for handwritten digit recognition based on global, local, structural and statistical features. The authors proposed image representation by global features: density, the center of gravity, the second-order geometrical moments, and the number of transitions, Hu's invariant moments, Zernike moments, skew, horizontal and vertical projections histograms, profile features: contour and skeleton of the image and coefficients of Ridgelet transform. In total, there were 101 features. When only one set of features was used, the highest classification accuracy was obtained when only global features were used. In that case, the error rate was 6.19%. By combining all features, the error rate dropped to 3.38%.

Other widely used features are extracted from the frequency domain. For transforming the image from the spatial to the frequency domain, a discrete cosine transform (DCT) or a discrete wavelet transform (DWT) can be used. The discrete wavelet transform generates four sub-images that represent high and low frequencies along with different directions. In ([Aider et al., 2018](#)), the support vector machine was trained with a combination of different sub-images generated by the wavelet transform as features. Based on the reported results, usage of just high frequencies leads to the lowest classification accuracy with an error rate of 29.23%. On the other hand, with only low frequency components for features, the error rate was minimal, 1.24%. It was an expected result since low frequencies are less sensitive to different writing styles.

Features extracted from the frequency domain were used for training the support vector machine in ([El qacimy et al., 2014](#)). The DCT generates a matrix of frequency coefficients of the same size as the original image in the spatial domain. In ([El qacimy et al., 2014](#)), the importance of the number of the coefficients used for handwritten digit recognition was tested as well as the order in which they are put into the input feature vector. Handwritten digits are of size 28x28, so in total, there are 784 coefficients. The presented results show that if the 100 coefficients from the upper left corner (low frequencies) were put in the feature vector, read in the zig-zag order, the error rate was 1.29% while the 256 low-frequency coefficients read sequentially gave the error rate of 1.34%. The lowest error rate was obtained when the DCT was applied to non-overlapping blocks of size 7x7 and 10 coefficients from each of these 16 non-overlapping blocks (the input image was 28x28 pixels, 16 blocks of 7x7 in total) were put into the feature

vector in the zig-zag order. The error rate, in that case, was 1.24%. The classification was done by the support vector machine.

The authors in (Das et al., 2012) proposed the support vector machine classification with a combination of 84 quad-tree based hierarchically derived longest-run (QTLR) features and 200 modular principal component analysis features. The QTLR features are topological features and are often used for optical character recognition. The problem with the QTLR features is that they are not good for capturing global statistical information which is why the authors proposed a combination with the MPCA method. The MPCA is a statistical method that has been commonly used for dimension reduction by finding the best variance of data. The proposed method tested on the MNIST dataset obtained an error rate of 1.60%.

## Artificial neural networks

Artificial neural networks (ANN) along with the SVM with the appropriate feature set have been giving the lowest error rate for the handwritten digit recognition problem. An ANN consists of the input and output layers, with one or more hidden layers between them. Each layer has a number of nodes as parameters. The number of nodes in the input layer is equal to the number of features that describe the images in the training set while the output layer has one node for each class. The nodes in the hidden layer are chosen by the user and the value of each node is obtained as the weighted sum of the node values from the previous layer. The simple ANN is presented in Fig. 6.

In (Ciresan et al., 2010), five different ANN architectures were tested for handwritten digit recognition. The MNIST dataset was used for testing, but the training set was expanded by deformed images. Digit images were transformed using affine and elastic transformations. The proposed networks had from 2 to 5 hidden layers and the number of neurons in each layer was between 500 and 2500. The last tested network had 9 hidden layers with 1000 nodes in each of them. The highest classification accuracy was achieved with the network with 5 hidden layers and the error rate was 0.32%.

The committee of 25 neural networks for handwritten digit recognition was presented in (Meier et al., 2011). Each network had one hidden layer with 800 nodes. The proposed method was tested on the original MNIST

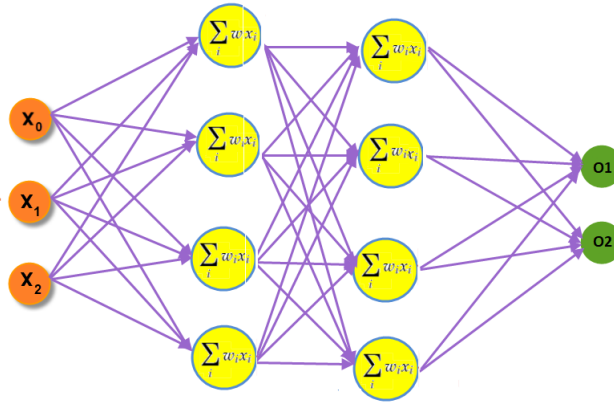


Figure 6 – An example of the ANN architecture

Рис. 5 – Пример архитектуры ANN

Слика 5 – Пример архитектура вештачке неуралне мреже

dataset as well as on preprocessed and deformatted images. The average error rate was 0.38%.

For digital image classification, the best type of neural networks to use are convolutional neural networks.

### Convolutional neural networks

Convolutional neural networks (CNN) are a specific type of neural networks that are especially useful for classification problems where the correlation between inputs is strong and should be considered when processing them. Examples of such problems are voice and image classifications. The CNN were introduced decades ago, but the lack of computational power and large datasets for training held further research until several years ago.

The accuracy of the CNN highly depends on their architecture and hyperparameters. Finding the optimal configuration for the CNN represents a hard optimization problem (Tuba et al., 2021). It is equivalent to the problem of the traveling salesman, and there is no deterministic algorithm that can produce the result in a reasonable amount of time. Solving such problems are of interest for decades and various effective techniques have been proposed so far. One group of algorithms named swarm intelligence algorithms was among the most efficient stochastic methods for tackling hard optimization problems. In recent years, the classification error produced

by the CNN is reduced by optimizing the network by swarm intelligence algorithms.

In (Tuba & Tuba, 2021), a simple optimization of the CNN for the MNIST dataset was proposed. The firefly algorithm was proposed for finding hyperparameters such as kernel size, padding and the number of feature maps while other hyperparameters were fixed. The architecture was based on LeNet-5. The optimized CNN has an error rate 0.84% while the original LeNet-5 has 1.06%.

Adaptation of the particle swarm optimization (PSO) for the CNN optimization was proposed in (Li et al., 2019). The quantum behaved particle swarm optimization with binary encoding (BQPSO) was used to find the optimal values for the kernel size, the number of feature maps, stride as well as padding of the convolutional layers, while the considered hyperparameters of the pooling layer are padding, kernel size, and type of pooling layers. Finally, fully connected layers were optimized by finding the values for the number of neurons. The architecture of the network was also variable of the BQPSO, i.e. the number and order of different layers. Besides the original MNIST dataset, in (Li et al., 2019) the complex MNIST dataset was used. That dataset contains rotated digits and background images. The error rate was 0.99%.

The hyperparameters of the CNN were tuned by the artificial bee colony optimization (ABC) in (Zhu et al., 2019). The ABC algorithm was used to set values for 13 CNN hyperparameters. Some of the hyperparameters that were considered are the number, type and order of layers, kernel sizes, learning rate, batch size and dropout probability. The classification error of the proposed CNN tested on the MNIST dataset was 0.74%.

In (Bacanin et al., 2020), the hybridized approach was proposed for finding the optimal configuration for the CNN. The monarch butterfly optimization algorithm (MBO) was improved by adding operators from the artificial bee colony algorithm and the hybridized method (MBO-ABC) was used for finding the optimal CNN. In total, 5 hyperparameters of the convolutional layer, 6 hyperparameters of the pooling layer and 3 general hyperparameters were tuned. The original MBO generated a CNN with the error rate of 0.36% while the hybridized approach, the MBO-ABC, 0.34%.

## Summary

Besides considering the classification accuracy or the error rate as a metric for comparing different classifiers, it is important to mention training and classification time. In general, the training time with the same features depends on the classifier and chosen features. More features will most likely improve the classification accuracy but it will also extend training and classification time since for each input instance it is necessary to find those features. In general, the kNN does not have the training time, but the classification time is long and depends on the number of instances in the training set. All other classifiers have the training time where linear classifiers, the naive Bayesian and the SVM with the linear kernel are similar, while for the ANN and the CNN training time depends on the network architecture. However, the ANN and the CNN usually do not have a feature extraction step which results in reduced classification time once the network is trained. All methods presented in this survey are used for offline handwritten digit recognition and training time is less important than classification accuracy. With today's technology, the calculation of all mentioned features is very fast and usually not considered an important measure of the handwritten digit recognition methods.

Table 1 summarize the results presented in this survey.

## Conclusion

Handwritten digit recognition is a problem of a great practical importance. In the last several decades, multiple different classification methods were proposed for this problem and for the purpose of testing the benchmark dataset was created, the MNIST dataset. Nowadays, the classification error on the MNIST dataset is less than 0.5% when using convolutional neural networks, while other classifiers can also achieve classification accuracy above 99%. Since it is a well studied problem, the MNIST dataset is now not just a dataset for the handwritten digit recognition problem, but it is also used as a standard benchmark dataset for testing new classifiers. Based on the detailed literature review, it can be concluded that the CNN are the state-of-the-art method for image classification, while deep artificial neural networks and the optimized SVM can give comparable results. Linear classifiers are not suitable for this problem.



Table 1 – Classification errors for the MNIST dataset with different classifiers  
 Таблица 1 – Ошибки классификации в наборе данных MNIST с разными классификаторами  
 Табела 1 – Грешка класификације МНИСТ базе са различитим класификаторима

Method	Description	Error rate (%)
Linear classifier (Lecun et al., 1998)	NA	12.00
Set of linear classifiers (Lecun et al., 1998)	One against all method for binary classifier	7.60
Linear SVM (Ebrahimzadeh & Jampour, 2014)	Histogram of oriented gradients	2.75
kNN (Grover & Toghi, 2020)	Euclidean distance, expanded training dataset	2.27
kNN (Babu et al., 2014)	Structural and statistical features	1.58
kNN	Rotation invariant LBP	10.19
Naive Bayesian (Wang & Zhang, 2020)	Binarization	18.60
Naive Bayesian (Armstrong, 2019)	Structural features, histogram of oriented gradients, and image moments	10.03
SVM (Tuba et al., 2016)	Histograms of projections to four different axes	4.40
SVM (Gattal et al., 2014)	Density, center of gravity, the second order geometrical moments and number of transitions, Hu's invariant moments, Zernike moments, skew, horizontal and vertical projections histograms	6.19
	added contour and skeleton of the image and coefficients of the Ridgelet transform	3.38
SVM (Patel & Kalyani, 2016)	Fringe distance map	2.86
	Inverse fringe distance map	2.28
SVM (Das et al., 2012)	84 quad-tree based hierarchically derived longest-run and 200 modular principal component analysis features	1.60
SVM (Aider et al., 2018)	High frequencies coefficients of the DWT	29.23
	Low frequencies of the DWT	1.24

Method	Description	Error rate (%)
SVM (El qacimy et al., 2014)	Upper left corner of the DCT coefficients in the zig-zag order	1.29
	256 DCT coefficients, upper left corner, sequentially read	1.34
	10 DCT coefficients from each of 16 non-overlapping blocks of size 7x7 in the zig-zag order	1.24
ANN (Ciresan et al., 2010)	Raw images	0.32
ANN (Meier et al., 2011)	Raw images	0.38
CNN (Tuba & Tuba, 2021)	Raw images, FA	0.84
CNN (Li et al., 2019)	Raw images, PSO	0.99
CNN (Zhu et al., 2019)	Raw images, ABC	0.74
CNN (Bacanin et al., 2020)	Raw images, MBO	0.34

## References

- Aider, M.A., Hammouche, K. & Gaceb, D. 2018. Recognition of handwritten characters based on wavelet transform and SVM classifier. *The International Arab Journal of Information Technology*, 15(6), pp. 1082–1087 [online]. Available at: <https://iajit.org/portal/PDF/November%202018,%20No.%206/10880.pdf> [Accessed: 1 March 2022].
- Armstrong, S. 2019. Naive-bayesian-mnist. *GitHub*. [online]. Available at: <https://github.com/sjnarmstrong/naive-bayesian-mnist> [Accessed: 10 March 2022].
- Babu, U.R., Chintla, A.K. & Venkateswarlu, Y. 2014. Handwritten digit recognition using structural, statistical features and k-nearest neighbor classifier. *International Journal of Information Engineering and Electronic Business (IJIEEB)*, 6(1), pp. 62–68. Available at: <https://doi.org/10.5815/ijieeb.2014.01.07>.
- Bacanin, N., Bezdán, T., Tuba, E., Strumberger, I. & Tuba, M. 2020. Monarch Butterfly Optimization Based Convolutional Neural Network Design. *Mathematics*, 8(6, art.ID:936), pp. 1–32. Available at: <https://doi.org/10.3390/math8060936>.
- Bhagya Shree, S. & Sheshadri, H. 2018. Diagnosis of Alzheimer's disease using Naive Bayesian Classifier. *Neural Computing and Applications*, 29, pp. 123–132. Available at: <https://doi.org/10.1007/s00521-016-2416-3>.
- Ciresan, D.C., Meier, U., Gambardella, L.M. & Schmidhuber, J. 2010. Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition. *Neural Computation*, 22(12), pp. 3207–3220. Available at: [https://doi.org/10.1162/NECO\\_a\\_00052](https://doi.org/10.1162/NECO_a_00052).

Cortes, C. & Vapnik, V. 1995. Support-vector networks. *Machine Learning*, 20(3), pp. 273–297. Available at: <https://doi.org/10.1007/BF00994018>.

Das, N., Reddy, J.M., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M. & Basu, D.K. 2012. A statistical–topological feature combination for recognition of hand-written numerals. *Applied Soft Computing*, 12(8), pp. 2486–2495. Available at: <https://doi.org/10.1016/j.asoc.2012.03.039>.

Deotte, C. 2018. How to score 97%, 98%, 99%, and 100%. *Kaggle*. [online]. Available at: <https://www.kaggle.com/c/digit-recognizer/discussion/61480> [Accessed: 10 March 2022].

Ebrahimzadeh, R. & Jampour, M. 2014. Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM. *International Journal of Computer Applications*, 104(9), pp. 10–13. Available at: <https://doi.org/10.5120/18229-9167>.

El qacimy, B., Ait kerroum, M. & Hammouch, A. 2014. Handwritten digit recognition based on DCT features and SVM classifier. In: *2014 Second World Conference on Complex Systems (WCCS)*. Agadir, Morocco, pp. 13–16, November 10-12. Available at: <https://doi.org/10.1109/ICoCS.2014.7060935>.

Esmaili, I., Dabanloo, N.J. & Vali, M. 2016. Automatic classification of speech dysfluencies in continuous speech based on similarity measures and morphological image processing tools. *Biomedical Signal Processing and Control*, 23, pp. 104–114. Available at: <https://doi.org/10.1016/j.bspc.2015.08.006>.

Gattal, A., Chibani, Y., Djeddi, C. & Siddiqi, I. 2014. Improving Isolated Digit Recognition Using a Combination of Multiple Features. In: *2014 14th International Conference on Frontiers in Handwriting Recognition*. Hersonissos, Greece, pp. 446–451, September 01-04. Available at: <https://doi.org/10.1109/ICFHR.2014.81>.

Grover, D. & Toghi, B. 2020. MNIST dataset classification utilizing k-NN classifier with modified sliding-window metric. In: *Arai, K. & Kapoor, S. (Eds.) Advances in Computer Vision. CVC 2019. Advances in Intelligent Systems and Computing*. 944, pp.583–591. Cham, Switzerland: Springer. Available at: [https://doi.org/10.1007/978-3-030-17798-0\\_47](https://doi.org/10.1007/978-3-030-17798-0_47).

Huang, H.Y. & Lin, C.J. 2016. Linear and Kernel Classification: When to Use Which? In: *Proceedings of the 2016 SIAM International Conference on Data Mining (SDM)*. Miami, Florida, USA, pp. 216–224, May 5-7. Available at: <https://doi.org/10.1137/1.9781611974348.25>.

Ilmi, N., Budi, W.T.A. & Nur, R.K. 2016. Handwriting digit recognition using local binary pattern variance and K-Nearest Neighbor classification. In: *2016 4th International Conference on Information and Communication Technology (ICoICT)*. Bandung, Indonesia, pp. 1–5, May 25-27. Available at: <https://doi.org/10.1109/ICoICT.2016.7571937>.

Kang, C., Huo, Y., Xin, L., Tian, B. & Yu, B. 2019. Feature selection and tumor classification for microarray data using relaxed Lasso and generalized multi-class support vector machine. *Journal of Theoretical Biology*, 463, pp. 77–91. Available at: <https://doi.org/10.1016/j.jtbi.2018.12.010>.



Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278–2324. Available at: <https://doi.org/10.1109/5.726791>.

Li, Y., Xiao, J., Chen, Y. & Jiao, L. 2019. Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification. *Neurocomputing*, 362, pp. 156–165. Available at: <https://doi.org/10.1016/j.neucom.2019.07.026>.

Lyu, H.K., Park, C.H., Han, D.H., Kwak, S.W. & Choi, B. 2018. Orchard Free Space and Center Line Estimation Using Naive Bayesian Classifier for Unmanned Ground Self-Driving Vehicle. *Symmetry*, 10(9, art.ID:355), pp. 1–14. Available at: <https://doi.org/10.3390/sym10090355>.

Meier, U., Cireşan, D.C., Gambardella, L.M. & Schmidhuber, J. 2011. Better Digit Recognition with a Committee of Simple Neural Nets. In: *2011 International Conference on Document Analysis and Recognition*. Beijing, China, pp.1250–1254, September 18-21. Available at: <https://doi.org/10.1109/ICDAR.2011.252>.

Nosseir, A. & Roshdy, R. 2018. Extraction of Egyptian License Plate Numbers and Characters Using SURF and Cross Correlation. In: *ICSIE '18: Proceedings of the 7th International Conference on Software and Information Engineering*. Cairo, Egypt, pp.48–55, May 02-04. Available at: <https://doi.org/10.1145/3220267.3220276>.

Patel, A. & Kalyani, T. 2016. Support Vector Machine with Inverse Fringe as Feature for MNIST Dataset. In: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. Bhimavaram, India, pp.123–126, February 27-28. Available at: <https://doi.org/10.1109/IACC.2016.32>.

Sethy, P.K., Barpanda, N.K., Rath, A.K. & Behera, S.K. 2020. Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175, art.number:105527. Available at: <https://doi.org/10.1016/j.compag.2020.105527>.

Sivaram, M., Laxmi, L.E., Pustokhina, I.V., Pustokhin, D.A., Elhoseny, M., Joshi, G.P. & Shankar, K. 2020. An optimal least square support vector machine based earnings prediction of blockchain financial products. *IEEE Access*, 8, pp. 120321–120330. Available at: <https://doi.org/10.1109/ACCESS.2020.3005808>.

Tuba, E., Bačanin, N., Strumberger, I. & Tuba, M. 2021. Convolutional Neural Networks Hyperparameters Tuning. In: *Pap, E. (Eds.) Artificial Intelligence: Theory and Applications. Studies in Computational Intelligence*. 973, pp.65–84. Cham, Switzerland: Springer. Available at: [https://doi.org/10.1007/978-3-030-72711-6\\_4](https://doi.org/10.1007/978-3-030-72711-6_4).

Tuba, E. & Tuba, I. 2021. Swarm Intelligence Algorithms for Convolutional Neural Networks. In: *2nd Workshop on Evolutionary and Population-based Optimization*. Online Event, pp.1–6, November 30. Available at: [https://wepo2021.aisylab.com/papers/wepo2021\\_paper\\_4.pdf](https://wepo2021.aisylab.com/papers/wepo2021_paper_4.pdf) [Accessed: 10 March 2022].

Tuba, E., Tuba, M. & Simian, D. 2016. Handwritten digit recognition by support vector machine optimized by Bat algorithm. In: *WSCG '2016: short communications proceedings: The 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2016 in co-operation with EUROGRAPHICS*. Plzen, Czech Republic: University of West Bohemia, pp.369-376, May 30-June 03. Available at: <http://hdl.handle.net/11025/29725> [Accessed: 10 March 2022].

Wang, H., Zheng, B., Yoon, S.W. & Ko, H.S. 2018. A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research*, 267(2), pp. 687–699. Available at: <https://doi.org/10.1016/j.ejor.2017.12.001>.

Wang, K. & Zhang, H. 2020. A Novel Naive Bayesian Approach to Inference with Applications to the MNIST Handwritten Digit Classification. In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA, pp.1354-1358, December 16-18. Available at: <https://doi.org/10.1109/CSCI51800.2020.00252>.

Yuan, G.X., Ho, C.H. & Lin, C.J. 2012. Recent Advances of Large-Scale Linear Classification. *Proceedings of the IEEE*, 100(9), pp. 2584–2603. Available at: <https://doi.org/10.1109/JPROC.2012.2188013>.

Zhai, Z., Xu, Z., Zhou, X., Wang, L. & Zhang, J. 2015. Recognition of hazard grade for cotton blind stinkbug based on Naive Bayesian classifier. *Transactions of the Chinese Society of Agricultural Engineering*, 31(1), pp. 204–211 [online]. Available at: <http://www.tcsae.org/nygcxben/article/abstract/20150128> [Accessed: 1 March 2022].

Zhu, W., Yeh, W., Chen, J., Chen, D., Li, A. & Lin, Y. 2019. Evolutionary Convolutional Neural Networks Using ABC. In: *ICMLC '19: Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. Zhuhai, China, pp.156–162, February 22-24. Available at: <https://doi.org/10.1145/3318299.3318301>.

---

Методы классификации при распознавании рукописных цифр: обзор

Ира М. Туба, Уна М. Туба, Младен Дж. Веинович, **корресподент**  
 Университет «Сингидунум», факультет информатики и  
 вычислительной техники, г. Белград, Республика Сербия

РУБРИКА ГРНТИ: 28.23.37 Нейронные сети  
 ВИД СТАТЬИ: обзорная статья

**Резюме:**

**Введение/цель:** В данной статье представлен обзор методов распознавания рукописных цифр, протестированных на наборе данных MNIST.

**Методы:** В данной статье анализируются, синтезируются и сравниваются разработки различных классификаторов, применяемых к задаче распознавания рукописных цифр, от линейных классификаторов до сверточных нейронных сетей.

**Результаты:** Точность классификации распознавания рукописных цифр, протестированная на наборе данных MNIST при использовании обучающих и тестовых наборов, теперь превышает 99,5%, а наиболее успешным методом является сверточная нейронная сеть.

**Выводы:** Распознавание рукописных цифр является проблемой во многих ситуациях реальной жизни. Точное распознавание различных стилей почерка, в частности цифр, изучается уже десятилетиями. В данной связи в статье обобщенно описаны достигнутые ранее результаты. Наилучшие результаты были достигнуты с помощью сверточных нейронных сетей, в то время как худшими методами оказались линейные классификаторы. Сверточные нейронные сети дают лучшие результаты, если набор данных расширяется с помощью дополнения данных.

**Ключевые слова:** распознавание рукописных цифр, классификация изображений, машина опорных векторов, глубокие нейронные сети, сверточные нейронные сети, оптимизация гиперпараметров, роевой интеллект, MNIST.

---

Методу класификације за препознавање руком писаних цифара: преглед

Ира М. Туба, Уна М. Туба, Младен Ђ. Веиновић, **аутор за преписку**

Универзитет Сингидунум, Факултет за информатику и рачунарство, Београд, Република Србија

ОБЛАСТ: математика, рачунарске науке

ВРСТА ЧЛАНКА: пегледни рад

**Сажетак:**

**Увод/циљ:** У раду је представљен преглед метода за препознавање руком писаних цифара тестираних на МНИСТ скупу података.

**Метод:** Рад анализира, синтетише и упоређује развој различитих класификатора примењених за препознавање руком писаних цифара, од линеарних класификатора до конволуцијских неуронских мрежа.

**Резултати:** Тачност класификације за препознавање руком писаних цифара тестирана на скуп података МНИСТ док се користи скуп за тренирање и тестирање је сада већа него 99,5%. Најуспешнија метода је конволуцијска неуронска мрежа.

**Закључак:** Тачно препознавање различитих стилова рукописа, конкретно ци- фара, проучавано је деценијама, а у раду су сумирани постигнути резултати. Најбољи су постигну- ти са конволуцијским неуронским мрежама, док су најлошије методе линеарни класификатори. Конволуцијске неуронске мреже дају боље резултате ако је скуп података проширен методом аугментације.

**Кључне речи:** препознавање руком писаних цифара, класификација слика, машина потпорних вектора, дубоке неуронске мреже, конволуцијске неуронске мреже, оптимизација хиперпараметара, интелигенција роја, МНИСТ.

Paper received on / Дата получения работы / Датум пријема чланка: 11.03.2022.  
Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 27.01.2023.

Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 29.01.2023.

© 2023 The Authors. Published by Vojnotehnički glasnik / Military Technical Courier (<http://vtg.mod.gov.rs>, <http://втр.мо.унп.срб>). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2023 Авторы. Опубликовано в "Военно-технический вестник / Vojnotehnički glasnik / Military Technical Courier" (<http://vtg.mod.gov.rs>, <http://втр.мо.унп.срб>). Данная статья в открытом доступе и распространяется в соответствии с лицензией "Creative Commons" (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2023 Аутори. Објавио Војнотехнички гласник / Vojnotehnički glasnik / Military Technical Courier (<http://vtg.mod.gov.rs>, <http://втр.мо.унп.срб>). Ово је чланак отвореног приступа и дистрибуира се у складу са Creative Commons лиценцом (<http://creativecommons.org/licenses/by/3.0/rs/>).

