Recognition of Handwritten Numerals Using Support Vector Machine

SOUVIK BANERJEE, SOUVIK ROY , PIJUS KANTI DHARA, SHAWNA SAHA, SOUMENDU DAS, PRANATI RAKSHIT
CSE Dept., JIS College Of Engineering

***Abstract*—Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data. Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning). In recent years, the recognition of printed and handwritten mathematical expressions has received an increasing amount of attention in pattern recognition research. The diversity of approaches to recognize mathematical expressions and the lack of a commercially viable system, however, indicate that there is still much research to be done in this area.**

***Index Terms*—Numeral Recognition, Machine learning, Support Vector Machine, training, supervised learning**

I. INTRODUCTION

Pattern recognition is one of the major challenges in statistics framework. Its primary goal is to extract efficient feature and accurately classify the patterns into categories. A well-known and vital application in this field is the handwritten digit classification and recognition where digits from 0 to 9 have to be assigned into one of the 10 classes using some classification method. The variety of digits to be recognized, the variations in handwriting, different notations and several other issues make this a complex problem. But these complexities do not prevent the system from having a wide applicability to real world problems and associated commercial uptake. It is very difficult to achieve 100% accuracy as pattern distortion, presence of unwanted objects and noise or disoriented patterns affect the performance of a recognition system. There are basically two approaches for handwritten digits classification and recognition- Online and Offline. Online character recognition involves the identification of character while they are being written which is captured by special hardware (e.g. smart pen or touchpad). In case of off-line character recognition, characters are written on paper with ordinary pen and converted into digital images using a camera or scanner and then those images are used for classification. In this paper, offline approach of digit classification has been followed.

II. LITERATURE REVIEW

Lihong Zheng and Xiangjian et al. [1] authors of the paper "**Number Plate Recognition Based on Support Vector Machine**" discuss on number plate recognition. It uses Support Vector Machine (SVM) algorithm to recognize numbers. The project starts from collecting a set of numbers from number plates. Every collected number is then recognized by using SVM. Before the recognition can be done using SVM, the SVM algorithm is first trained to by some known samples of numbers. The recognition result is achieved by finding the maximum value between the outputs of SVMs.

Bradley Schafer and Serestina Viriri et al. [2] authors of the paper **"Off-line Signature Verification System"** discuss on a topic called An Off-Line Signature Verification System which was published in year 2009. This journal is studied because it employs a basic image processing method. The signature goes through three main process which is preprocessing, feature extraction and finally classification.

Amritpal Kaur and Madhavi Arora et al. [3] authors of the paper **"Neural network based Numerical digits Recognization using NNT in Matlab"** discuss on number recognition which was based on a system that recognizes a english numeral, given by the user, which is already trained on the features of the numbers to be recognized using NNT (Neural network toolbox). The system has a neural network as its core, which is first trained on a database. The training of the neural network extracts the features of the English numbers and stores in the database. The next phase of the system is to recognize the number given by the user. The features of the number given by the user are extracted and compared with the feature database and the recognized number is displayed.

Stuti Asthana, Farha Haneef and Rakesh K Bhujade et al. [4] author of the paper **"Handwritten multi-script numeral recognition using artificial neuralnetworks"** proposed a Multilayer feed-forward back-propagation algorithm using two hidden layer was used. Network was trained to learn its behavior by adjusting the connection strengths on every iteration. The resultant of each presented training pattern was calculated to identify the minima on the error surface for each training pattern.

Yoshihiro Kojima, Hiroshi Yamamoto, Toshiyuki Kohda, Shigeo Sakaue, Susumu Maruno, Yasuharu Shimeki, Kazutaka Kawakami, Mikio Mizatani et al. [5] authors of the paper **"Recognition of Handwritten Numeric Characters Using Neural Networks Designed on Approximate Reasoning Architecture"** developed a handwritten numeric character recognition system with Neural networks designed on Approximate Reasoning Architecture (NARA) and obtained a correct answer rate of 95.41%, an error rate of 0.20% and a rejection rate of 4.38% of handwritten character images. NARA which consists of a classifier of input data, several sub-neural networks and an integrator of the outputs of sub-neural networks can realize a stable recognition of tremendous variation of handwritten character images.

Luiz S. Oliveira, Robert Sabourin, Ching Y. et al. [6] authors of the paper **“Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy”** propose a modular system to recognize handwritten numerical strings. It uses a segmentation-based recognition approach and a Recognition and Verification strategy. The approach combines the outputs from different levels such as segmentation, recognition, and post-processing in a probabilistic model. A new verification scheme which contains two verifiers to deal with the problems of over-segmentation and under-segmentation is presented. A new feature set is also introduced to feed the over-segmentation verifier. A postprocessor based on a deterministic automaton is used and the global decision module makes an accept/reject decision. Finally, experimental results on two databases are presented: numerical amounts on Brazilian bank checks and NIST SD19. The latter aims at validating the concept of modular system and shows the robustness of the system using a well-known database.

El-Sayed M. El-Alfy et al. [7] author of the paper **“A Hierarchical GMDH-Based Polynomial Neural Network for Handwritten Numeral Recognition Using Topological Features”** propose a multiclass hierarchical abductive learning classifier and apply it to improve the recognition rate of handwritten numerals while reduce the dimensionality of the feature space. For handwritten recognition, there are ten classes. Using 9 binary GMDH-based neural network models structured in a hierarchy has led to improving balance factor of the data set for each classifier and improving the classification of handwritten numerals. It also has the advantage of removing the need to resolve classification ties that exist in other forms of combining a number of classifiers to solve a multiclass classification problem whether using one-versus-all or one-versus-one approaches.

Nyssa Aragon, William Lane, Fan Zhang et al. [8] authors of the paper **“Classification of Hand-Written Numeric Digits”** design a specific hand-written recognition application that will read numeric digits which are written on a tablet or mobile device.The application has also a feature of traning itself with users specific set of data that will improve the predictions of digits written by those users.

Akash Choudhary et al. [9] author of the paper **“Hand-written English numerical recognition system using neural network”** states an algorithm for recognition of hand-written English numeral. The digits are classified into two groups, one group comprises of blobs with/without stems and the other digits with stems only. The blobs are identified based on a new concept called morphological region filling technique. This eliminates the issue of finding the size of blobs and their structuring elements. This method completely eliminates the complex process of recognition of horizontal or vertical lines.

Kai Labusch and Erhardt Barth et al. [10] author of the paper “**Simple Method for High-Performance Digit Recognition Based on Sparse Coding**” propose a method of feature extraction for digit recognition that is inspired by vision research: a sparse-coding strategy and a local maximum operation. They first employ the unsupervised Sparsenet algorithm to learn a basis for representing patches of handwritten digit images. They then use this basis to extract local coefficients. In a second step , they apply a local maximum operation in order to implement local shift invariance. Finally, they train a Support-Vector-Machine on the resulting feature vectors and obtain state-of-the-art classification performance in the digit recognition task defined by the MNIST benchmark. They compare the different classification performances obtained with sparse coding, Gabor wavelets, and principle component analysis. They conclude that the learning of a sparse representation of local image patches combined with a local maximum operation for feature extraction can significantly improve recognition performance.

Subhransu Maji and Jitendra Malik et al. [11] author of the paper **“Fast and Accurate Digit Classification”** explore the use of certain image features, blockwise histograms of local orien-tations, used in many current object recognition algorithms, for the task of hand-written digit recognition. Existing approaches find that polynomial kernel SVMs trained on raw pixels achieve state of the art performance. However such kernel SVM approaches are impractical as they have a huge complexity at runtime. They demonstrate that with improved features a low complexity classifier, in particular an additive-kernel SVM, can achieve state of the art performance. Their approach achieves an error of 0.79% on the MNIST dataset and 3.4% error on the USPS dataset, while running at speeds comparable to the fastest algorithms on these datasets which are based on multilayer neural networks and are significantly faster and easier to train.

Vineet Singh and Sunil Pranit Lal et al. [12] author of the paper **“Digit recognition using single layer neural network with principal component analysis”** presents an approach to digit recognition using single layer neural network classifier with Principal Component Analysis (PCA). The proposed model in this paper aims to reduce the features to reduce computation requirements and successfully classify the digit into 10 categories (0 to 9). The system designed consists of backward propagation (BP) neural network and is trained and tested on the MNIST dataset of handwritten digit. The proposed system was able to obtain 98.39% accuracy. PCA is used for feature extraction to curtail the computational and training time and at the same time produce high accuracy. It was clearly observed that the training time is reduced by up to 80% depending on the number of principal component selected.

III. MNIST DATASET

MNIST Handwritten Digits dataset has been used in our model validation. The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images. The database is also widely used for training and testing in the field of machine learning. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset. The original black and white (bi-level) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

TABLE I. MNIST Dataset

Class	**Number of samples**	
	Training Samples	***Test Samples***
0	5923	980
1	6746	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009
Total	60000	10000

For training of the digit recognition system, we have used 1000 samples of each class and for testing we have used first 500 samples of training set for each class.

IV. THEORY

A. Feature Extraction

Feature Extraction is used to extract relevant features for recognition of characters. Features should be easily computed, robust, insensitive to various distortions and variations in the images, and rotationally invariant. Two kinds of features are used in pattern recognition problems. One kind of features has clear physical meaning, such as geometric or structural and statistical features. Another kind of features has no physical meaning. We call these features mapping features. The advantage of physical features is that they need not deal with irrelevant features. The advantage of the mapping features is that they make classification easier because clear boundaries will be obtained between classes but increasing the computational complexity. Feature selection is to select the best subset from the input space. Its ultimate goal is to select the optimal features subset that can achieve the highest accuracy results. While feature extraction is applied in the situation when no physical features can be obtained. Most of feature selection algorithms involve a combinatorial search through the whole space. Usually, heuristic methods, such as hill climbing, have to be adopted, because the size of input space is exponential in the number of features. Other methods divide the feature space into several subspaces which can be searched easily.

First features are computed and extracted and then most relevant features are selected to construct feature vector which is used for recognition. The computation of features is based on statistical, structural, directional, moment, transformation like approaches. Feature extraction methods for handwritten character recognition are based on two types of features: Structural and Statistical. Structural features are based on topological and geometric properties of the character. Examples of structural features are number of horizontal lines, number of vertical lines, number of endpoints, number of cross points, horizontal curves at top or bottom etc.

The statistical features are derived from the statistical distributions of pixels. These features can be easily detected as compared to structural features. Statistical features are not affected too much by noise or distortions as compared to statistical features. A number of techniques are used for statistical feature extraction; some of these are: zoning, projection histograms, crossings and distances, n-tuples.

B. HOG Feature Descriptor

In this work **Histogram of Oriented Gradients** (HOG) feature descriptor is used. Histogram of Oriented Gradient (HOG) was first proposed by Dalal and Triggs [13] for human body detection but it is now one of the successful and popular used descriptors in computer vision and pattern recognition. HOG counts occurrences of gradient orientation in part of an image hence it is an appearance descriptor. HOG divides the input image into small square cells (here we used 4×4) and then computes the histogram of gradient directions or edge directions based on the central differences. For improve accuracy, the local histograms have been normalized based on the contrast and this is the reason that HOG is stable on illumination variation. It is a fast descriptor in compare to the SIFT and LBP due to the simple computations, it has been also shown that HOG features are successful descriptor for detection. In the HOG feature descriptor, the distributions (histograms) of directions of gradients (oriented gradients) are used as features. Gradients (x and y derivatives) of an image are useful because the magnitude of gradients is large around edges and corners (regions of abrupt intensity changes) and it is known that edges and corners pack in a lot more information about object shape than flat regions.

To calculate a HOG descriptor, we need to first calculate the horizontal and vertical gradients; after all, we want to calculate the histogram of gradients. This is easily achieved by filtering the image with the following kernels.

$$K_{GX} = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad K_{GY} = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Then the magnitude and direction of gradient are calculated using the following formula:

Gradient Magnitude, $M_{XY} = \sqrt{G_X^2 + G_Y^2}$

Gradient Direction, $\theta = \tan^{-1} \frac{G_Y}{G_X}$

At every pixel, the gradient has a magnitude and a direction. The magnitude of gradient at a pixel is the maximum of the magnitude of gradients of the three channels, and the angle is the angle corresponding to the maximum gradient. Then the image is divided into 4x4 cells and a histogram of gradients is calculated for each 4×4 cells. The cell size totally depends on the developer on the basis of trial and error method. The histogram is essentially a vector (or an array) of 9 bins (numbers) corresponding to angles 0, 20, 40, 60 … 160.

The next step is to create a histogram of gradients in these 4×4 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40 … 160.

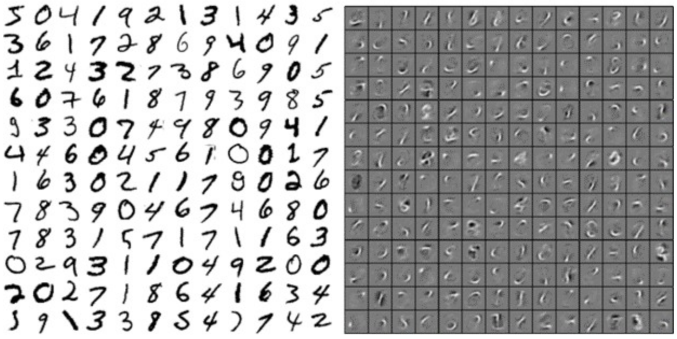

Fig. 1. HOG Features for some handwritten digits.

C. Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems. SVMs are based on the idea of finding a hyper-plane that best divides a dataset into two classes, as shown in Figure 2.

A linear support vector machine is composed of a set of given support vectors z and a set of weights w. The computation for the output of a given SVM with N support vectors $Z_1, Z_2, \ldots, Z_N$ and weights $W_1, W_2, \ldots, W_N$ is then given by:

$$F(x) = \sum_{i=1}^{N} w_i \langle z_i, x \rangle + b$$

where b is a weight vector known as bias. A decision function is then applied to transform this output in a binary decision. Usually, sign of the output is used, so that outputs greater than zero are taken as a class and outputs lesser than zero are taken as the other.

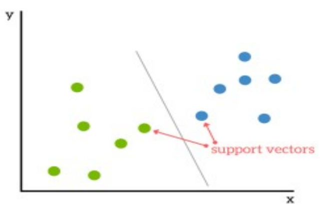

Fig. 2. Example of Support Vector Machine

Initially the SVM classifiers were proposed for binary classification. Generally multiclass SVM is implemented by combining binary SVMs either by one-versus-one or one-versus-all approach. In this paper we focus on the first approach. The one-versus-one method constructs $\frac{N(N-1)}{2}$ SVMs, considering all binary combinations of N classes. To decide the output of the multiclass SVM, a voting scheme is used in which outputs of all the binary SVMs are decided first and then the class which receives the most decisions is declared as the winner of the decision process.

V. IMPLEMENTATION

This section of the paper discuss about the implementation of proposed Handwritten Digit Recognition System. This implementation part can be broadly categorized into three steps:

A. Preprocessing

The preprocessing operations are classical operations in image processing, their objective is to clean and prepare the image for the other succeeding steps of a pattern recognition system. Here this step attempts to eliminate some unnecessary information and reduce the size of input images. As mentioned earlier, the dataset we have used, provides derived database from the original MNIST dataset and separates the training set into ten files for class 0-9, we didn't have to put a lot of effort in preprocessing part. In this work, preprocessing focuses on the binarization of the grayscale images. So, first those training images are read from the file. Then those grayscale images are converted to Black & White images using Ostu's [14] global image thresholding method.

B. Feature Extraction

There are several feature extraction techniques and the classification accuracy of a recognition system depends on the right selection of feature extraction technique. As it was mentioned earlier, HOG based feature extraction scheme was used in this work. Every binary image of size 28×28 pixel is used for HOG feature extraction. We have used a *cellsize* of 4×4 pixel. Each block consists of 2×2 cells. The blocks are 50% overlapped with the succeeding block. So, for an image of size 28×28 pixel, there total $36\ (6 \times 6)$ blocks. In our method we have taken the bin size of 9. So, for each image we have obtained a HOG feature vector of size 1296 (36×4×9). After obtaining the feature vector of each training image, we prepared the final feature vector by concatenating the individual feature vector of each training image. The size of the final feature vector is 10000×1296.

C. Training and Classification

The training stage is the process of training the handwritten digit recognition system with a set of input and target dataset. Input is represented using the feature vector of the training set and target is represented by class labels. We have used Support Vector Machine (SVM) as the classifier. So, the SVM was trained using the final feature vector and respective class labels.

Classification comes after the training process. When the classifier is trained, it is ready for classifying the test samples. The classifier returns a class label for the test sample presented to the classifier. In our case, we have prepared another feature vector with the test dataset in same manner as discussed in *Feature Extraction*. Then the feature vector was presented to the trained SVM for classification.

VI. RESULT

In this paper a handwritten digit classification and recognition system based on HOG feature descriptor and SVM as a classifier has been implemented successfully using the MATLAB 2015(a) software package. Table II shows the classification accuracy percentage of all the digits or classes. We have obtained an accuracy rate of 98.64% in this approach.

Table II. Confusion Matrix by with multiclass SVM with linear kernel

Class Names	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	99.4	0.2	0.4	0	0	0	0	0	0	0
1	0	99.8	0	0	0	0	0	0.2	0	0
2	0	0.4	99.2	0	0	0	0	0	0	0.4
3	0.2	0	0.6	97.4	0	0.4	0	0.4	0.6	0.4
4	0	0	0	0	98.2	0	0	0	0	1.8
5	0	0.2	0.2	0	0	98.8	0.6	0	0.2	0
6	0	0	0.2	0	0.6	0	99.2	0	0	0
7	0	0.6	0.2	0	0	0	0	98.4	0	0.8
8	0	0.2	0.2	0	0	0.2	0.4	0	98.4	0.6
9	0.2	0.2	0.2	0	0.8	0	0	1	0	97.6

We have achieved 98.64% accuracy rate which is comparable with other techniques reported in [15] where K-nearest neighbor has 91.6% accuracy, K-means clustered modified k-nearest classifier (KMA-MKNN) obtained 93.64% accuracy. Our proposed method also performed better than the method proposed in [16] which obtained 97.94% accuracy.

VII. CONCLUSION AND FUTURE SCOPE

In this paper, the proposed approach for handwritten digits recognition used HOG feature descriptor with linear multiclass SVM classifier. The performance of various classification methods still depend greatly on the general characteristics of the data to be classified. The exact relationship between the data to be classified and the performance of various classification methods still remains to be discovered. Thus far, there has been no classification method that works best on any given problem. There have been various problems to the current classification methods we use today. To determine the best classification method for a certain dataset we still use trial and error to find the best performance.

Though we have obtained quite high accuracy rate, still there are some flaws to be fixed. Misclassification rate of our proposed technique is 1.36%. From the confusion matrix shown in Table II, it is notable that the most confusions happened between classes of ‘1’, ‘7’ and ‘3’, ‘8’ and ‘7’, ‘9’ which make sense because upper section of the number of ‘7’ is similar to ‘1’ and ‘9’ and also the middle section of ‘8’ is similar to ‘3’. This leads to the further development of this proposed handwritten digit recognition system in near future. This work can also be extended to develop an offline handwritten mathematical expression evaluation system.

VIII. COPYRIGHT FORMS

You must submit the IEEE Electronic Copyright Form (ECF) as described in your author-kit message. THIS FORM MUST BE SUBMITTED IN ORDER TO PUBLISH YOUR PAPER.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put applicable sponsor acknowledgments here; DO NOT place them on the first page of your paper or as a footnote.

REFERENCES

[1] Lihong Zheng and Xiangjian He, “Number Plate Recognition based on Support Vector Machines”, University of Technology, Sydney, 2007.

[2] B.Schafer and S.Viriri - An Off-Line Signature Verification System‖, 2009, (ICSIPA- 2009), pp.95-100

[3] Amritpal Kaur and Madhavi Arora, “Neural network based Numerical digits Recognition using NNT in Matlab”, Global Institute of Engineering and Technology, Amritsar.

[4] Stuti Asthana, Farha Haneef and Rakesh K Bhujade, “Handwritten Multiscript Numeral Recognition using Artificial Neural Networks”, Int. J. of Soft Computing and Engineering , ISSN: 2231-2307, Volume-1, Issue-1, March 2011.

[5] Yoshihiro Kojima, Hiroshi Yamamoto, Toshiyuki Kohda, Shigeo Sakaue, Susumu Maruno, Yasuharu Shimeki, Kazutaka Kawakami, and Mikio Mizutani. Recognition of handwritten numeric characters using neural networks designed on approximate reasoning architecture. In Proc. Ijcnn-93, International Joint Conference on Neural Networks, Nagoya volume III, pages 2161--2164, Piscataway, NJ, 1993. JNNS, IEEE Service Center.

[6] Ching Y. Suen, Flávio Bortolozzi, Robert Sabourin, Luiz S. Oliveira, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy", IEEE

Transactions on Pattern Analysis & Machine Intelligence, vol. 24, no. , pp. 1438-1454, November 2002.

[7] El-Sayed M. El-Alfy, Offline recognition of handwritten numeral characters with polynomial neural networks using topological features, Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence, May 31-June 02, 2010, Ottawa, Canada

[8] Aragon, Nyssa, William Lane, and Fan Zhang. "Classification of Hand-Written Numeric Digits." Stanford.edu. N.p., 12 Dec. 2013. Web. 29 Apr. 2017

[9] Akash Choudhary, "Hand-Written English Numeral Recognition System Using Neural Network"

[10] K. Labusch, E. Barth and T. Martinetz, "Simple Method for High-Performance Digit Recognition Based on Sparse Coding," in *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1985-1989, Nov. 2008. doi: 10.1109/TNN.2008.2005830

[11] Subhransu Maji, Jitendra Malik, "Fast and Accurate Digit Classification", EECS Department, UCB, Tech. Rep. UCB/EECS-2009-159, Nov. 2009

[12] V. Singh and S. P. Lal, "Digit recognition using single layer neural network with principal component analysis," *Asia-Pacific World Congress on Computer Science and Engineering*, Nadi, 2014, pp. 1-7. doi: 10.1109/APWCCSE.2014.7053842

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005

[14] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics. 1979, 9(1): 62-66.

[15] M. Narasimha Murty, V. Susheela Devi. An Application: Handwritten Digit Recognition, ISBN: 978-0-85729-494- 4, Springer, 2011,pp. 247-254

[16] R. F. P. Neves, A. N. G. Lopes Filho, C. A. B. Mello and C. Zanchettin, "A SVM based off-line handwritten digit recognizer," 2011 IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, 2011, pp. 510-515.