

## Benchmark of Different Gradient Descents in OCR

*Marjan Kuchaki Rafsanjani, Masoud Pourshaban*

*Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran  
E-mail: kuchaki@uk.ac.ir*

**Abstract:** *In this paper we implement six different learning algorithms in Optical Character Recognition (OCR) problem and achieve the criteria of end-time, number of iterations, train-set performance, test-set performance, validate-set performance and overall performance of these methods and compare them. Finally, we show the advantages and disadvantages of each method.*

**Keywords:** *Learning algorithm, neural network, Optical Character Recognition (OCR), pattern recognition.*

### 1. Introduction

Optical Character Recognition (OCR) is a field of research in pattern recognition, artificial intelligence and machine vision. It refers to the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text [1].

Besides, the extensive applications of character recognition in recognizing the numbers on bank checks [2], car plate numbers [3], etc., have caused varieties of new systems, such as the OCR system to be developed [4]. In character recognition references, several different methods, such as the decision tree [5, 6], fuzzy set theory [7-10], artificial neural networks [11, 12], support vector machines [13, 14], hidden Markov model [15-18] or any other significant hybrid of these methods [19, 20] are exploited.

Feature extraction methods for handwritten characters and digits have been mainly based on two types of features: (a) statistically derived from the statistical distribution of points and (b) structural. The most common statistical features used for character representation are: (a) zoning, where the character is divided into several zones and the features are extracted from the densities in each zone [21], or

from measuring the direction of the character contour by computing histograms of chain codes in each zone [22], (b) projections [23] and (c) crossings, that consist of the number of transitions from foreground to background pixels along horizontal and vertical lines and distances, that rely on the calculation of the distance of the first foreground pixel detected from the upper/lower (left/right) boundaries of the image along vertical (horizontal) lines [24].

The document image analysis has been accentuated as one of the main research issues for about three decades and several methods for recognition of Indian [25, 26], Thai [18], Chinese [27-30] and Japanese [31-33] documents have been proposed. Also, numerous researches have been realized on recognition of Farsi and Arabic texts [1, 4, 34-39]

The process of OCR algorithms is shown in Fig. 1.

This paper is organized as follows: In Section 2 the learning method that we use in this paper is briefly described. In Section 3 we implement these methods and getting the results, compare them and finally conclude the paper by Section 4.

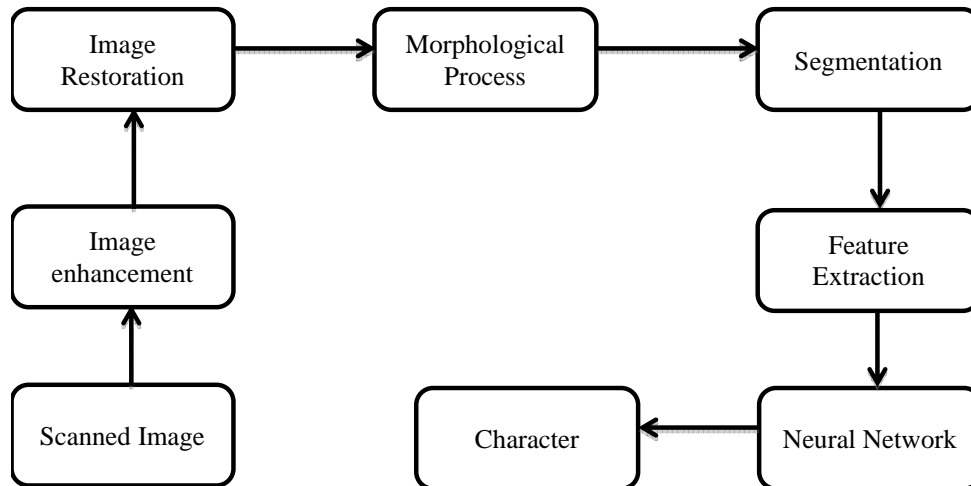


Fig. 1. Stages of OCR that a neural network uses

## 2. Preliminaries

A number of advanced algorithms have been proposed in neural networks learning so far. Methods, such as conjugate gradients [40], Quasi-Newton are considered popular choices for training feed-forward neural networks. These algorithms attempt to use second derivative-related information to accelerate the learning process [41]. In the following subsections the methods are briefly discussed.

### 2.1. Fletcher-Powell Conjugate Gradient (CGF)

All the conjugate gradient algorithms start by searching in the steepest descent direction (negative of the gradient) at the first iteration:

$$(1) \quad p_0 = -g_0.$$

A line search is then performed to determine the optimal distance to move along the current search direction:

$$(2) \quad x_{k+1} = x_k + \alpha_k p_k.$$

Then the next search direction is determined so that it is conjugate to the previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction:

$$(3) \quad p_k = -g_k + \beta_k p_{k-1}.$$

The various versions of the conjugate gradient algorithm are distinguished by the manner, in which the constant  $\beta_k$  is computed. For Fletcher-Reeves method given in [42], the procedure updated is

$$(4) \quad \beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}.$$

This is the ratio of the norm squared of the current gradient to the norm squared of the previous gradient [42-43].

## 2.2. Polak-Ribière Conjugate Gradient (CGP)

Polak and Ribière proposed another version of the conjugate gradient algorithm. As with Fletcher-Reeves algorithm, the search direction at each iteration is

$$(5) \quad p_k = -g_k + \beta_k p_{k-1}.$$

For the Polak-Ribière update, the constant  $\beta_k$  is

$$(6) \quad \beta_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}}.$$

This is the inner product of the previous change in the gradient with the current gradient divided by the norm squared of the previous gradient [42-43].

## 2.3. Conjugate Gradient with Powell/Beale restarts (CGB)

For all conjugate gradient algorithms, the search direction is periodically reset to the negative of the gradient. The standard reset point occurs when the number of iterations is equal to the number of network parameters (weights and biases), but there are other reset methods that can improve the training efficiency.

Such a reset method was proposed by Powell [44], based on an earlier version proposed by Beale [45]. This technique restarts if there is very little orthogonality left between the current gradient and the previous gradient. This is tested by equation

$$(7) \quad |g_{k-1}^T g_k| \geq 0.2 \|g_k\|^2.$$

If this condition is satisfied, the search direction is reset to the negative of the gradient.

#### 2.4. Scaled Conjugate Gradient (SCG)

Each of the conjugate gradient algorithms requires a line search at each iteration. This line search is computationally expensive, because it requires the network response to all training inputs to be computed several times for each search. SCG developed by Moller [46], was designed to avoid the time-consuming line search. This algorithm combines the model-trust region approach with the conjugate gradient approach.

#### 2.5. One Step Secant (OSS)

The OSS method is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm does not store the complete Hessian matrix; it assumes that at each iteration, the previous Hessian was the identity matrix. This has the additional advantage that the new search direction can be calculated without computing the matrix inverse [47].

#### 2.6. Resilient Back-propagation (RP)

The most popular training algorithm of this category is the batch BP [48]. It is a first-order method that minimizes the error function by updating the weights using the steepest descent method [41].

The purpose of the Resilient back **propagation** (Rprop) training algorithm is to eliminate the harmful effects of the magnitudes of the partial derivatives.

Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor whenever the derivative with respect to that weight changes the sign from the previous iteration. If the derivative is zero, the update value remains the same. When the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for several iterations, the magnitude of the weight change increases [49].

### 3. Experimental results

We have above implemented the Gradient algorithm on char74K Database [50] with MATLAB (Version 2012a) 10 times. The properties of the artificial neural network and the computer we used are listed in Table 1. You can see the results and the comparison among the methods in Figs 2-8.

Table 1. Properties of our system and the neural network

| Train set size | Test set size | Validate set size | Hidden layer | No nodes in each hidden layer | Output | Performance | Computer                |
|----------------|---------------|-------------------|--------------|-------------------------------|--------|-------------|-------------------------|
| 10850          | 2325          | 2325              | 2            | 30                            | 62     | $10^{-5}$   | Apple MacBook Pro MD101 |

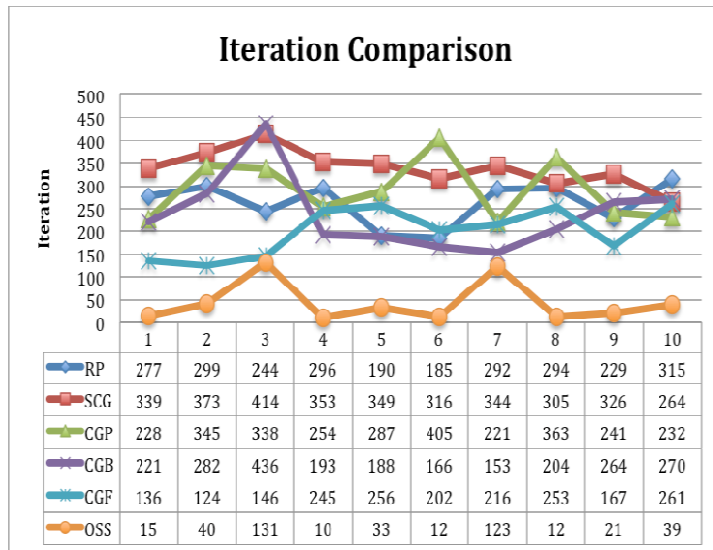


Fig. 2. Time comparisons

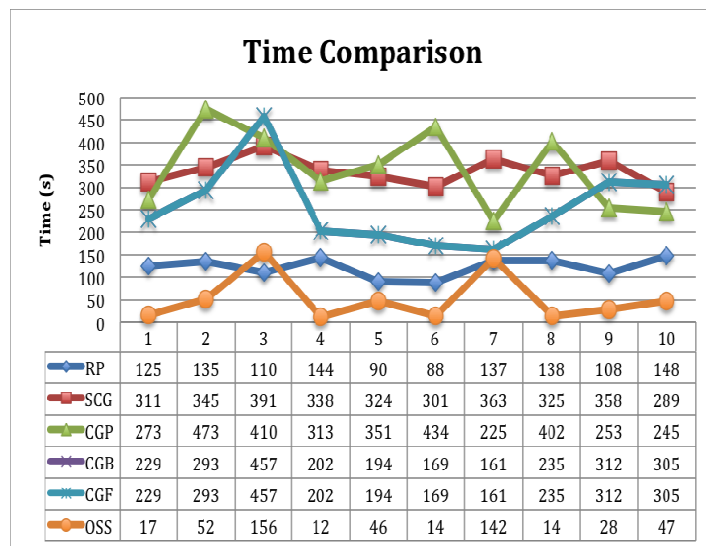


Fig. 3. Iterations comparisons

Fig. 2 proves that the OSS learning method is better than the other methods with respect to the time comparison. Except for two runs, OSS has a great end time.

At the next position the RP method is located. RP has the best stability in running; it means that RP method has a little difference end time compared to the others. The upper end time of RP is 144 and the lower end time is 88. On the other hand, CGF has the worst stability that means that if you want to use this method, you must get several runs.

In Fig. 3 we show the number of iterations at each run of the methods. As expected for the low end time of OSS method, OSS has a low number of iterations,

but if you look carefully, you will see that RP has a higher number of iterations, it means more iterations for one second in RP than OSS or another method. If we divide the iterations in time, we can see that at each run RP has a better result than the other methods.



Fig. 4. Train-performance comparisons

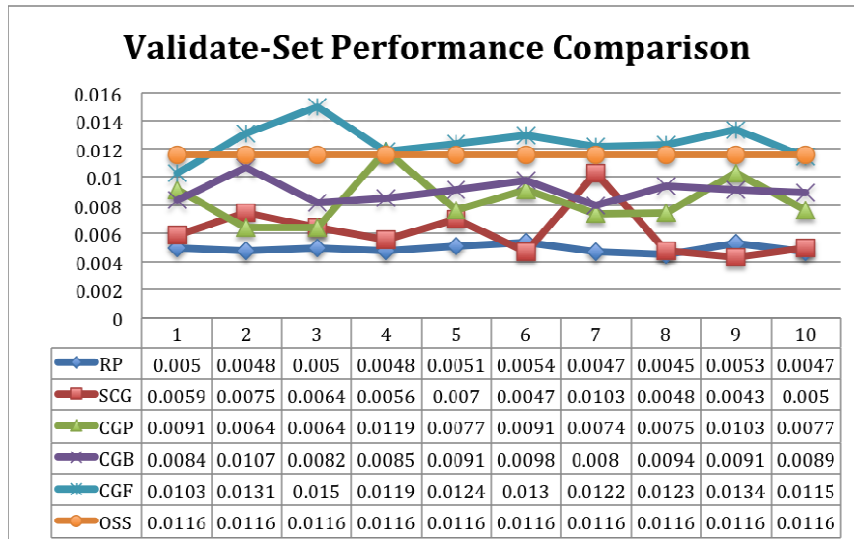


Fig. 5. Performance comparisons

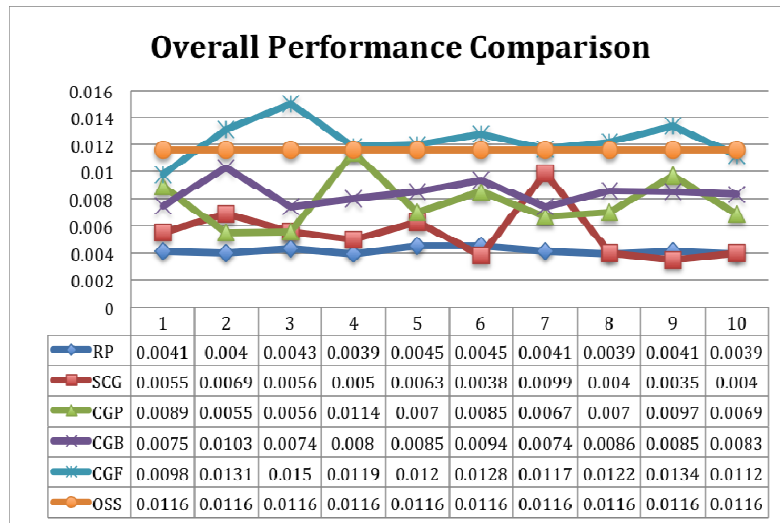


Fig. 6. Test-performance comparisons

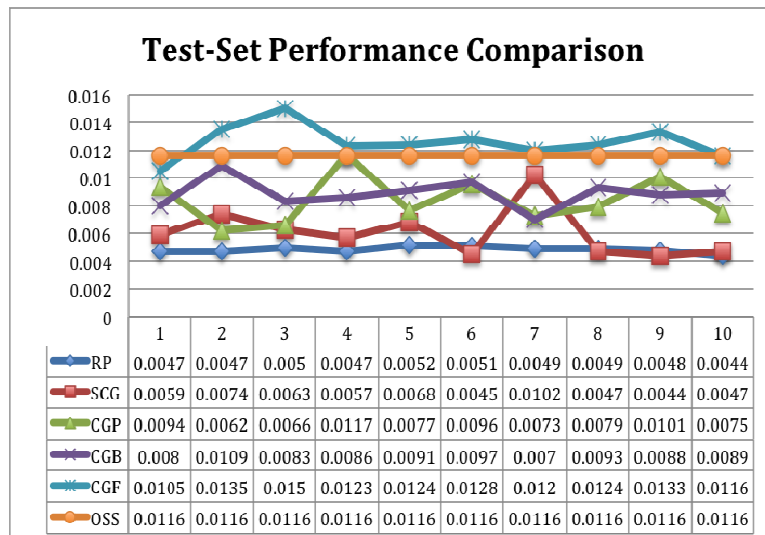


Fig. 7. Validate-performance comparisons

Performance means difference between the data that are given to ANN and the output data that the neural network produces. It is obvious that if we have a low performance number it means that the difference is low and we have a good neural network that has good predictive ability and low generalization error.

In Fig. 4 the train-set performance at each run of the methods is shown and again we see that the RP method has the low performance number among the others, so that this method has the best predictive ability and a low generalization error. At the next position SCG method is found. This method in two cases of running is better than RP, but it does not have the stability of RP.

In Fig. 5 the validate-set performance is shown, in Fig. 6 the test-set performance is presented, and finally in Fig. 7 the overall performance of the neural

network in each run of the methods is shown. As you can see, the results in Figs 5-7 are very similar to Fig. 4.

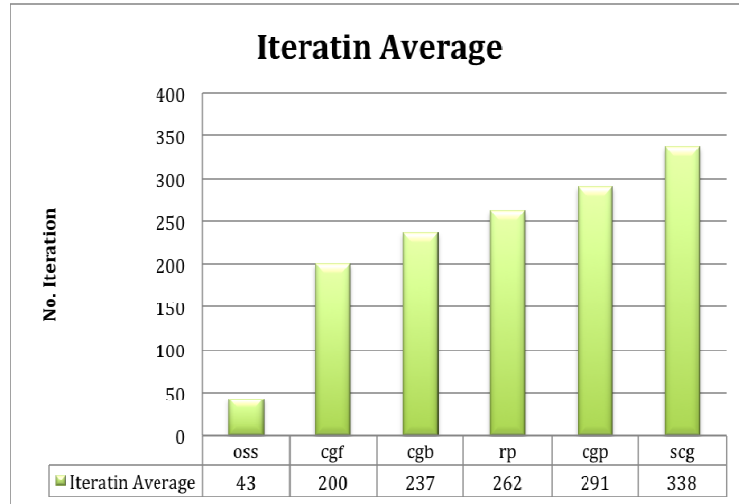


Fig. 8. Time average

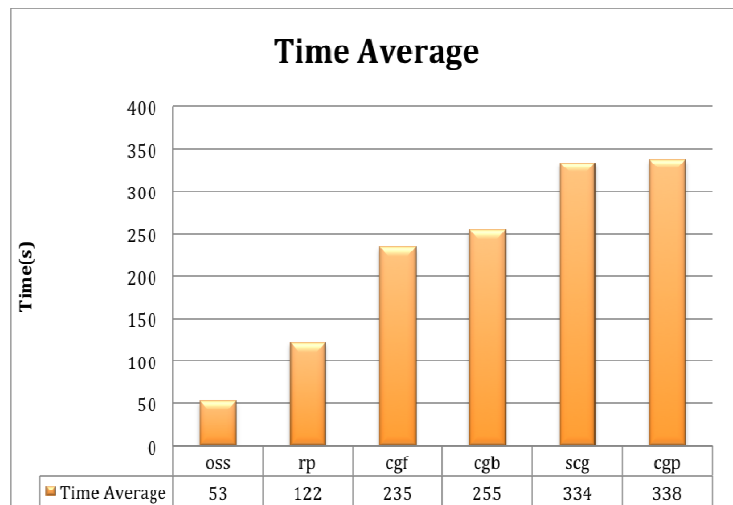


Fig. 9. Iteration average

In Fig. 8 we get the time average from 10 times run of the methods. As expected, OSS method with 53 end-time average has been accomplished earlier than the other methods. At the next position RP method is located with 122 end-time average, and so on. The significant point in this figure is the huge difference end-time of some methods, for example, the end-time difference of RP method and OSS method in Fig. 8 is 69, the difference end-time of RP method and CGF is 113, and the difference end-time of CGB method and SCG is 79.



Fig. 9 shows the average of the iterations number of the methods. The highest number of iterations means that the method makes an effort to decrease the difference between the output of the neural network and the target set.

For the speed test of the neural network we divide the number of iterations to time in order to specify how many iterations has got each method in one second. In average, OSS method has 0.81 iteration per second, RP method has 2.1 iterations per second, CGF method has 0.85 iteration per second, CGB method has 0.93 iteration per second, CGP method has 0.86 iteration per second, SCG method has 1.01 iteration per second.

In Fig. 10 the average of the train-set performance of each method is shown. Fig. 11 presents the average of the validate-set performance. Fig. 12 shows the average of the test-set performance and Fig. 13 presents the average of the overall performance of each method.

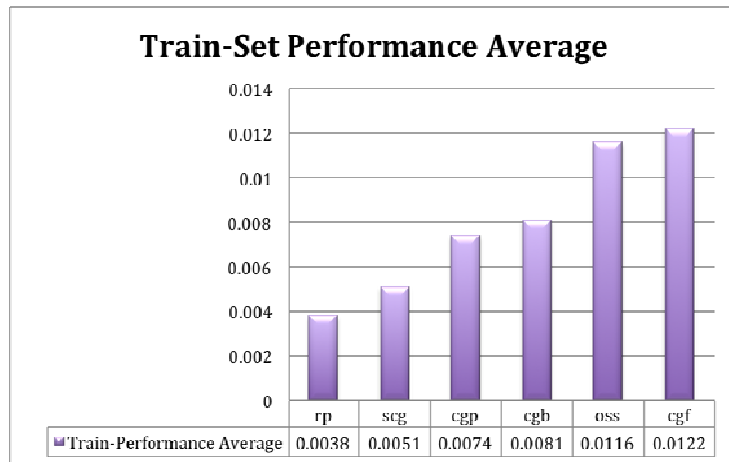


Fig. 10. Train-set performance average

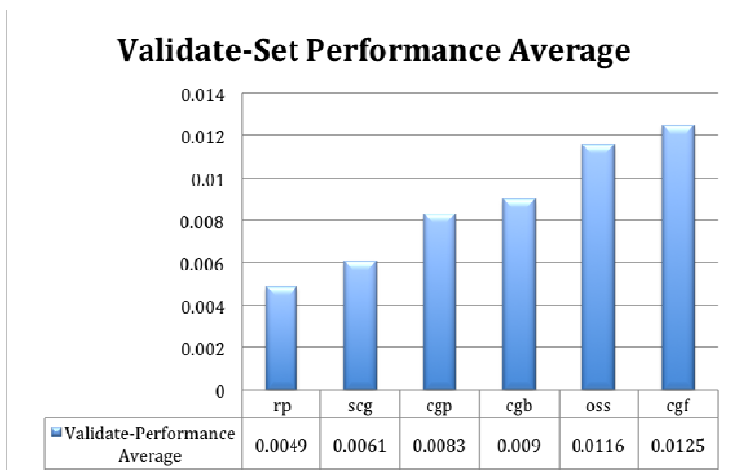


Fig. 11. Validate-set performance average

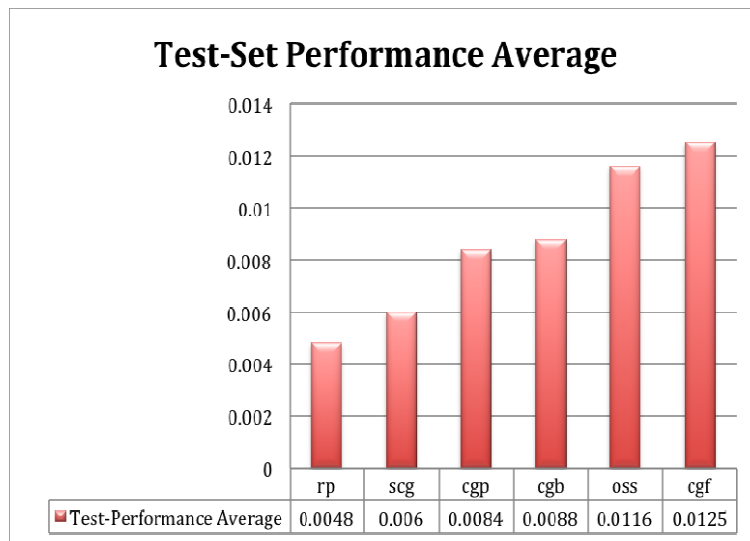


Fig. 12. Test-set performance average

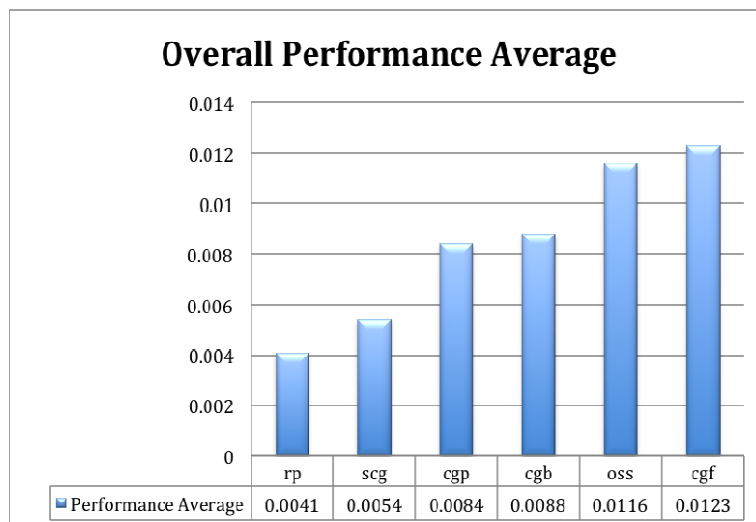


Fig. 13. Overall performance average

In all figures the RP method has the least difference between the input and output of the neural network. At the second place SCG method comes, CGP and CGB methods have almost similar results in performance and take the third and the fourth place.

OSS and CGF methods have similar results, but a relatively large difference compared to other methods and take the fifth and the sixth place.

## 4. Conclusion

In this paper we implement six different learning algorithms. OSS is good with respect to end-time but not so good in performance, RP method is good for OCR because of the low end-time and good performance. Besides, RP method has the best stability and a high number of iterations at each run, SCG method has good performance, but it does not have a good end-time, CGP has an average performance and it takes too long to end, CGP is average in both time and performance. CGB method has the worst performance and an average end-time.

## References

1. Broumandnia, A., J. Shanbehzadeh. Fast Zernike Wavelet Moments for Farsi Character Recognition. – *Image and Vision Computing*, Vol. **25**, 2007, 717-726.
2. Morita, M. E., E. Lethelier, A. E. Yacoubi, F. Bortolozzi, R. Sabourin, P. Pontifícia. Recognition of Handwritten Dates on Bankchecks Using an HMM Approach. – In: 13th Brazilian Symposium on Computer Graphics and Image Processing, Gramado, Brazil, 2000.
3. Pan, X., X. Ye, S. Zhang. A Hybrid Method for Robust Car Plate Character Recognition. – *Engineering Applications of Artificial Intelligence*, Vol. **18**, 2005, No 8, 963-972.
4. Al-Alawi, R. A Hybrid  $n$ -Tuple Neuro-Fuzzy Classifier for Handwritten Numerals Recognition. – In: IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 2004.
5. Amin, A. Recognition of Printed Arabic Text Based on Global Features and Decision Tree Learning Techniques. – *Pattern Recognition*, Vol. **33**, 2000, No 8, 1309-1323.
6. Nagabhushan, P., R. M. Pai. Modified Region Decomposition Method and Optimal Depth Decision Tree in the Recognition of Non-Uniform Sized Characters – An Experimentation with Kannada Characters. – *Pattern Recognition Letters*, Vol. **20**, 1999, No 14, 1467-1475.
7. Cheok, A. D., Z. Jian, E. S. Chng. Efficient Mobile Phone Chinese Optical Character Recognition Systems by Use of Heuristic Fuzzy Rules and Bigram Markov Language Models. – *Applied Soft Computing*, Vol. **8**, 2008, No 2, 1005-1017.
8. Chi, Z., M. Suters, H. Yan. Handwritten Digit Recognition Using Combined ID3-Derived Fuzzy Rules and Markov Chains. – *Pattern Recognition*, Vol. **29**, 1996, No 11, 1821-1833.
9. Hanmandlua, M., K. R. M. Mohanb, S. Chakrabortyc, S. Goyal d. Unconstrained Handwritten Character Recognition Based on Fuzzy Logic. – *Pattern Recognition*, Vol. **36**, 2003, 603-623.
10. Patil, P. M., T. R. Sontakke. Rotation, Scale and Translation Invariant Handwritten Devanagari Numeral Character Recognition Using General Fuzzy Neural Network. – *Pattern Recognition*, Vol. **40**, 2007, No 7, 2110-2117.
11. Goltsev, A., D. Rachkovskij. Combination of the Assembly Neural Network with a Perceptron for Recognition of Handwritten Digits Arranged in Numeral Strings. – *Pattern Recognition*, Vol. **38**, 2005, No 3, 315-322.
12. Lauer, F., C. Y. Suen, G. Bloch. A Trainable Feature Extractor for Handwritten Digit Recognition. – *Pattern Recognition*, Vol. **40**, 2007, No 6, 1816-1824.
13. Camastra, F. A SVM-Based Cursive Character Recognizer. – *Pattern Recognition*, Vol. **40**, 2007, No 12, 3721-3727.
14. Malon, C., S. Uchida, M. Suzuki. Mathematical Symbol Recognition with Support Vector Machines. – *Pattern Recognition Letters*, Vol. **29**, 2008, No 9, 1326-1332.
15. Amor, N. B., N. E. B. Amara. Multifont Arabic Character Recognition Using Hough Transform and Hidden Markov Models. – In: Fourth International Symposium on Image and Signal Processing and Analysis, Zagreb, Croatia, 2005.

16. Bunke, H., M. Roth, E. Schukat-Talamazzini. Off-Line Cursive Handwriting Recognition Using Hidden Markov Models. – Pattern Recognition, Vol. **28**, 1995, No 1, 399-1413.
17. Khorsheed, M. S. Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model. – Pattern Recognition Letters, Vol. **24**, 2003, No 14, 2235-2242.
18. Theeramunkong, T., C. Wongtapan. Off-Line Isolated Handwritten Thai OCR Using Island-Based Projection with  $n$ -Gram Model and Hidden Markov Models. – Information Processing and Management, Vol. **41**, 2005, No 1, 139-160.
19. Nagabhushan, P., R. M. Pai. Modified Region Decomposition Method and Optimal Depth Decision Tree in the Recognition of Non-Uniform Sized Characters – An Experimentation with Kannada Characters. – Pattern Recognition Letters, Vol. **20**, 1999, No 14, 1467-1475.
20. Ping, Z., C. Lihui. A Novel Feature Extraction Method and Hybrid Tree Classification for Handwritten Numeral Recognition. – Pattern Recognition Letters, Vol. **23**, 2002, No 1-3, 45-56.
21. Oliveira, L. S., F. Bortolozzi, C. Y. Suen. Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy. – IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. **24**, 2001, No 11, 1448-1456.
22. Mohiuddin, K. M., J. Mao. A Comprehensive Study of Different Classifiers for Hand-Printed Character Recognition. – Pattern Recognition, Practice IV, 1994, 437-448.
23. Koerich, A. L. Unconstrained Handwritten Character Recognition Using Different Classification Strategies. – In: International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR), 2003.
24. Kim, H., K. K. Kim, C. Y. Suen. Hybrid Schemes of Homogeneous and Heterogeneous Classifiers for Cursive Word Recognition. – In: Proceedings of Seventh International Workshop on Frontiers in Handwritten Recognition, Amsterdam, 2000, 433-442.
25. Al-Omari, F. A., O. Al-Jarrah. Handwritten Indian Numerals Recognition System Using Probabilistic Neural Networks. – Advanced Engineering Informatics, Vol. **18**, 2004, No 1, 9-16.
26. Pal, U., B. B. Chaudhuri. Indian Script Character Recognition: A Survey. – Pattern Recognition, Vol. **37**, 2004, No 9, 1887-1899.
27. Hung, K.-Y., R. W. P. Luk, D. S. Yeung, K. F. L. Chung, W. Shu. A Multiple Classifier Approach to Detect Chinese Character Recognition Errors. – Pattern Recognition, Vol. **38**, 2005, No 5, 723-738.
28. Li, Y.-X., C. L. Tan, X. Ding, C. Liu. Contextual Post-Processing Based on the Confusion Matrix in Offline Handwritten Chinese Script Recognition. – Pattern Recognition, Vol. **37**, 2004, No 9, 1901-1912.
29. Su, T.-H., T.-W. Zhang, D.-J. Guan, H.-J. Huang. Off-Line Recognition of Realistic Chinese Handwriting Using Segmentation-Free Strategy. – Pattern Recognition, Vol. **42**, 2009, No 1, 167-182.
30. Zhao, S., Z. Chi, P. Shi, H. Yan. Two-Stage Segmentation of Unconstrained Handwritten Chinese Characters. – Pattern Recognition, Vol. **36**, 2003, No 1, 145-156.
31. Itoh, N. Japanese Language Model Based on Bigrams and Its Application to On-Line Character Recognition. – Pattern Recognition, Vol. **28**, 1995, No 2, 135-141.
32. Kimura, F., T. Wakabayashi, S. Tsuruoka, Y. Miyake. Improvement of Handwritten Japanese Character Recognition Using Weighted Direction Code Histogram. – Pattern Recognition, Vol. **30**, 1997, No 8, 1329-1337.
33. Srihari, S. N., T. Hong, G. Srikantan. Machine-Printed Japanese Document Recognition. – Pattern Recognition, Vol. **30**, 1997, No 8, 1301-1313.
34. Al-Muhtaseb, H. A., S. A. Mahmoud, R. S. Qahwaji. Recognition of Off-Line Printed Arabic Text Using Hidden Markov Models. – Signal Processing, Vol. **88**, 2008, No 12, 2902-2912.
35. Azmi, R., E. Kabir. A New Segmentation Technique for Omnifont Farsi Text. – Pattern Recognition Letters, Vol. **22**, 2001, 97-104.
36. Dehghan, M., K. Faez, M. Ahmadi, M. Shridhar. Unconstrained Farsi Handwritten Word Recognition Using Fuzzy Vector Quantization and Hidden Markov Models. – Pattern Recognition Letters, 2001, 209-214.

37. Mozaffari, S., K. Faez, V. Märgner, H. El-Abed. Lexicon Reduction Using Dots for Off-Line Farsi/Arabic Handwritten Word Recognition. – *Pattern Recognition Letters*, Vol. **29**, 2008, No 6, 724-734.
38. Solimanpour, F., J. Sadri, C. Y. Suen. Standard Databases for Recognition of Handwritten Digits, Numerical Strings, Legal Amounts, Letters and Dates in Farsi Language. – In: 10th Int'l Workshop on Frontiers of Handwriting Recognition, La Baule, France, 2006.
39. Ziaratban, M., K. Faez, F. Faradji. Language-Based Feature Extraction Using Template-Matching in Farsi/Arabic Handwritten Numeral Recognition. – In: 9th Int'l Conference on Document Analysis and Recognition, Curitiba, Brazil, 2007.
40. Blumenstein, M., X. Y. Liu, B. Verma. A Modified Direction Feature for Cursive Character Recognition. – In: IEEE International Joint Conference on Neural Networks, Vol. **4**, 2000, 2983-2987.
41. Fitzgerald, J. A., F. Geiselbrechtinger, T. Kechadi. Application of Fuzzy Logic to Online Recognition of Handwritten Symbols. – In: Proceedings of Ninth International Workshop on Frontiers in Handwritten Recognition, 2004, 395-400.
42. Fletcher, R., C. M. Reeves. Function Minimization by Conjugate Gradients. – *Computer Journal*, Vol. **7**, 2004, 149-154.
43. Hagan, M. T., H. B. Demuth, M. H. Beale. *Neural Network Design*. Boston, MA, PWS Publishing, 1996.
44. Powell, M. J. D. Restart Procedures for the Conjugate Gradient Method. – *Mathematical Programming*, Vol. **12**, 1977, 241-254.
45. Beale, E. M. L. A Derivation of Conjugate Gradients. – In: F. A. Lootsma, Ed. *Numerical Methods for Nonlinear Optimization*. London, Academic Press, 1972.
46. Moller, M. F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. – *Neural Networks*, Vol. **6**, 1993, 525-533.
47. Battiti, R. First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method. – *Neural Computation*, Vol. **2**, 1992, No 4, 141-166.
48. Sexton, A. P., V. Sorge. Database-Driven Mathematical Character Recognition, Graphics Recognition, Algorithms and Applications (GREC). – In: *Lecture Notes in Computer Science (LNCS)*, Hong Kong, 2006, 206-217.
49. Riedmiller, M., H. Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. – In: *Proceedings of IEEE International Conference on Neural Networks*, 1993.
50. De Campos, T. E., B. R. Babu, M. Varma. Character Recognition in Natural Images. – In: *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal, February 2009.