

The Extended-Average Common Submatrix Similarity Measure with Application to Handwritten Character Images

Alessia AMELIO^{1*}, Darko BRODIĆ², Radmila JANKOVIĆ³

¹*DIMES, University of Calabria, Via Pietro Bucci 44, 87036 Rende (CS), Italy*

²*Technical Faculty in Bor, University of Belgrade, V.J. 12, 19210 Bor, Serbia*

³*Mathematical Institute of the Serbian Academy of Sciences and Arts*

Kneza Mihaila 36, 11000 Belgrade, Serbia

e-mail: aamelio@dimes.unical.it, dbrodic@tfbor.bg.ac.rs, rjankovic@mi.sanu.ac.rs

Received: December 2017; accepted: June 2018

Abstract. This paper introduces a new similarity measure derived from the Common Submatrix-based measures for comparing square matrices. The novelty is that the similarity between two matrices is computed as the average area of the largest sub-matrices exactly matching and being located at the same position in the two matrices. By contrast, in the original similarity measures, the largest sub-matrices can exactly or approximately match and be located at different positions. An experiment conducted on a subset of the MNIST and NIST datasets shows that the new similarity measure is very promising in retrieving relevant handwritten character images.

Key words: pattern matching, 2d array, similarity, image processing, characters recognition.

1. Introduction

A similarity measure takes into consideration two real problems. In the first one, a reference object is observed. In the second one, the other object is in the focus. Basically, it is evaluated how much it matches the first object (Goshtasby and Ardeshir, 2012). This object is usually called the target object. Hence, a similarity measure quantifies how similar are these two objects (reference and target). Also, this includes the matching of parts of the given object.

Basically, the use of a specific similarity metrics deeply depends on the problem which has to be solved. In particular, it is defined by the goal of a measurement process. The final goal is the similarity level which is derived by the matching between a target object and a reference object. However, the metrics heavily depend on the characteristics of the objects to be compared.

Typically, we can consider the following similarity metrics: (i) mean squares, (ii) normalized correlation, (iii) pattern intensity, (iv) mutual information, and (v) self-similarity matrix.

* Corresponding author.

Mean squares or Mean Square Error (MSE) uses the intensity values given by two objects. It is typically used as an evaluator in a one-dimensional problem. Basically, it evaluates the difference between the estimator and the reference by measuring the average of the squares of the errors, i.e. deviations between them. This fact implies that their values should be in the same range. In fact, MSE measures the level of estimator's quality. It is worth noting that it always receives a positive value, and if it is closer to zero, then the estimation is better. However, in another way, MSE can be interpreted as a second moment of the error. Accordingly, it includes the variance of the estimator as well as its bias. If we have an unbiased estimator, then the MSE represents the variance of the estimator. In such cases, MSE can be used as a good approximation of the difference between estimator and reference. Still, it suffers from the same disadvantages as the variance if an estimator includes heavily weighted outliers. Due to calculating the squared difference, large error elements appear. In this case, the MSE approach can be troublesome in some types of applications.

Normalized correlation calculates the correlation between the intensity values that characterize the two objects. In this way, it establishes a similarity measure between them. Basically, both objects are represented as two different vectors. Consequently, this method is used for the template matching, which finds the patterns matching in both objects. The normalized correlation can receive values in the range between $+1$ and -1 . Accordingly, a linear relationship between the objects is established. Hence, it also represents the two-dimensional version of the Pearson's correlation coefficient. The advantage of the method is its easy implementation and computer time effectiveness. However, the main disadvantage of the method is its limitation to the only linear problems.

Pattern intensity uses the differences between intensity values transformed by a function which is usually given as $1/(1+x)$. After that, these differences are squared and then summed at the end. These metrics are simultaneously increased when more samples are accessible as well as when the intensity values are close to each other. It represents its advantage over some other metrics methods.

Mutual information includes the metrics which are based on the information theory postulate. Basically, it measures the dependence between two objects represented by variables. In this way, it quantifies how much one variable tells about the other one. This measure gives the amount of information obtained by one variable through the other variable. The initial concept of mutual information is connected to the entropy. Accordingly, it can be considered to be a reduction in uncertainty about one variable given the knowledge of another variable. The obtained value is a dimensionless quantity with units in bits. Zero values of the mutual information mean that two variables are mutually independent. On the contrary, a high value of mutual information demonstrates a high reduction in uncertainty, while a low value reveals a small reduction in uncertainty.

Self-similarity matrix graphically represents similar sequences in a specific data series. The data series typically are feature vectors characterized by an N length. Then, it computes a square self-similarity matrix of area $N \times N$ showing pairwise similarities or dissimilarities between vectors. Basically, the self-similarity matrices match a series of features to show similarities and differences between those features in the series. The diagonal elements correlate to the self-comparison, giving zero values in the diagonal. On

the contrary, each element of the matrix is calculated according to a distance measure, such as the Euclidian distance (or some other) and the features. This allows the comparison of repetitive elements and segments. Higher scores are given to more similar elements (including repetitive elements), and lower or negative scores are characterized by dissimilar elements. If similar sequences representing repetitive elements exist, then diagonal stripes are shown in the plot.

At the end, it is worth noting that the choice of the right similarity measure depends on the objects that are compared.

The aforementioned measures as well as their extensions are used for the objects that are usually represented as bi-dimensional ones. Hence, these objects can be represented as matrices. In that sense, the similarity of two objects can be evaluated in two different circumstances: (i) the determination of the similarity between two matrices, and (ii) the determination of the similarity between a template matrix and a window in a larger matrix.

According to the previous categorization, many similarity measures have been proposed in the literature to evaluate the matching of the objects. Some of them are statistics-based: (i) Pearson-Correlation Coefficient (Pearson, 1896), (ii) Stochastic Sign Changed (Duda *et al.*, 2000), (iii) Median of Square Differences (Lan *et al.*, 1995), (iv) Intensity-Ratio Variance (Woods *et al.*, 1992), (v) Intensity-Mapping-Ratio Variance (Hill *et al.*, 1993), and (vi) Rank Distance (Goshtasby and Ardeshir, 2012). Also, some measures have geometrical properties, such as: (i) Tanimoto Measure (Theodoridis and Koutroumbas, 2009), and (ii) Normalized Square L_2 Norm (Evangelidis and Psarakis, 2008). Other measures are information-theoretic, such as: (i) Incremental Sign Distance (Kaneko *et al.*, 2002), (ii) Joint Entropy (Shannon, 1948), (iii) Kullback-Leibler Divergence (Kullback and Leibler, 1951), (iv) Exclusive F-Information (Rougon *et al.*, 2003), (v) Jensen-Shannon Divergence (Fuglede and Topsoe, 2004), and (vi) Arithmetic-Geometric Mean Divergence (Tourassi *et al.*, 2007).

Lately, the similarity between matrices has been measured by Common Submatrix approaches typically on image examples with the introduction of the baseline *Average Common Submatrix* measure (ACSM) (Amelio and Pizzuti, 2013, 2016), and its variant *Approximate Average Common Submatrix* (A-ACSM) (Amelio, 2016), which overcome the limitations of the previously introduced (dis)similarity measures. They compute the average area of the largest square sub-matrices matching in two images for measuring their similarity. These methods are based on the principle that two matrices are more similar to each other if they share larger patterns. The application of the Common Submatrix approaches is predominantly linked to the images. However, they can be used in any other area of interest in signal processing where data can be represented in the matrix format.

The main limitation of the baseline ACSM measure is its computational complexity, which is critical when its brute-force algorithm is applied. Although a tree indexing strategy has been introduced in ACSM for reducing the temporal cost of the brute-force algorithm, the construction procedure of the tree is quite demanding in time and space. On the other hand, the A-ACSM measure reduces the temporal cost of the brute-force algorithm of ACSM by omitting the comparison of a portion of elements at regular intervals during the match of the sub-matrices. Its main advantage is that it does not need any additional tree indexing data structure to be constructed and kept in memory. However, setting

the correct size of the interval is a critical aspect which may compromise the similarity computation.

In this paper, we propose an optimized version of the Common Submatrix approaches for measuring the similarity. The new measure only considers the match of those sub-matrices which are located at the same position in the two matrices. Also, the new measure exactly matches the sub-matrices because every element of the first sub-matrix is compared with its corresponding element of the second sub-matrix. Hence, any element is omitted from the comparison during the match. In this way, an improvement in time complexity of the baseline ACSM method is achieved by including all elements in the comparison of the sub-matrices, with no need of constructing a tree indexing data structure. An experiment is conducted on a subset of a 1000 images from the MNIST dataset and a subset of 210 images from the NIST dataset for the retrieval of handwritten character images. The obtained results show that the new similarity measure is very promising in terms of retrieval precision and execution time, when it is compared with other well-known (dis)similarity measures in the state-of-the-art.

The paper is organized as follows. Section 2 recalls the main concepts underlying the baseline ACSM measure from which the new similarity measure is derived. Section 3 presents the main features of the new similarity measure. Section 4 describes the experiment, i.e. the procedure for performance evaluation, the adopted datasets, and the competing methods. Section 5 shows the results in terms of retrieval precision and execution time. At the end, Section 6 draws a conclusion and outlines future work directions.

2. The ACSM Similarity Measure

ACSM is a suitable measure for computing the (dis)similarity between matrices, which can be represented as images (Amelio and Pizzuti, 2013). The ACSM measure is based on the concept that two matrices are similar to each other if the average area of their common sub-matrices is large. This quantifies the similarity between the two matrices. Then, the dissimilarity can be computed from the similarity value (Amelio and Pizzuti, 2013).

2.1. The Methodology

Let A and B be two square matrices respectively of area $N \times N$ and $M \times M$ defined on the same alphabet Σ . For each position (i, j) in A , ACSM finds the largest square sub-matrix exactly matching a square sub-matrix at some position (h, k) in B . An exact match implies checking all elements in the two sub-matrices and verifying if they are identical at the corresponding positions. The area of the largest common square sub-matrices is progressively summed. Finally, the average area is computed as the similarity value between A and B .

The ACSM similarity measure is defined as follows:

$$S_{\alpha}(A, B) = \sum_{i=1}^N \sum_{j=1}^N \frac{W(i, j)}{N^2}, \quad W(i, j) \geq \alpha, \quad (1)$$

where $W(i, j)$ is the area of the largest square sub-matrix at position (i, j) in A exactly matching a square sub-matrix at some position (h, k) in B , and α is a parameter setting the minimum allowed area of the square sub-matrices to be considered.

The ACSM similarity measure varies between 0 (any match occurs between A and B) and $\sum_{i=1}^N \sum_{j=1}^N \frac{\min\{i, j\}^2}{N^2}$ (when A and B are identical). In this last case, for each position (i, j) in A , the area of the largest square sub-matrix exactly matching inside B is the maximal area $\min\{i, j\}^2$ (Amelio and Pizzuti, 2013).

2.2. The ACSM Algorithm

Let $A(i, j)^k$ be the square sub-matrix of area $k \times k$ and bottom-right corner at position (i, j) in A . The procedure for computing the ACSM similarity measure is reported in Algorithm 1.

Algorithm 1 ACSM similarity measure.

Require: A, B, α

```

1:  $W_{tot} := 0, d := 0, i := 1$ 
2: while  $i \leq N$  do
3:    $j := 1$ 
4:   while  $j \leq N$  do
5:      $W(i, j) := 0, d := \min\{i, j\}, flag := true$ 
6:     while  $d \geq \sqrt{\alpha}$  and  $flag$  do
7:       if  $\mathcal{M}(A(i, j)^d, B)$  then
8:          $flag = false$ 
9:       else
10:         $d = d - 1$ 
11:       end if
12:     end while
13:     if  $\neg flag$  then
14:        $W(i, j) = d^2$ 
15:     end if
16:      $W_{tot} = W_{tot} + W(i, j), j = j + 1$ 
17:   end while
18:    $i = i + 1$ 
19: end while
20:  $S_{\alpha}(A, B) = W_{tot}/N^2$ 
21: return  $S_{\alpha}(A, B)$ 

```

The algorithm iterates over A (steps 2–4). For each position (i, j) in A , it finds the largest square sub-matrix of area bigger than or equal to α exactly matching inside B (steps 6–12). It is accomplished starting from the square sub-matrix of maximal area $\min\{i, j\}^2$ at position (i, j) in A (step 5). Its exact match is verified with a sub-matrix

$$A = \begin{vmatrix} 1 & 3 & 2 & 1 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 2 \end{vmatrix} \quad B = \begin{vmatrix} 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 1 \end{vmatrix}$$

Fig. 1. Sample matrices A and B of area 5×5 defined on the same alphabet $\Sigma = \{1, 2, 3\}$.

$$A = \begin{vmatrix} 1 & 3 & 2 & 1 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 2 \end{vmatrix} \quad B = \begin{vmatrix} 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 1 \end{vmatrix}$$

$$A(5,4)^4 = \begin{vmatrix} 2 & 1 & 3 & 2 \\ 1 & 1 & 3 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 3 & 1 & 2 \end{vmatrix} \quad A(5,4)^3 = \begin{vmatrix} 1 & 3 & 1 \\ 1 & 2 & 1 \\ 3 & 1 & 2 \end{vmatrix}$$

$$A(5,4)^2 = B(3,5)^2 = \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix}$$

Fig. 2. Extraction of the largest common square sub-matrix at position $(5, 4)$ in A . This is $A(5, 4)^2$ of area 2×2 exactly matching the sub-matrix $B(3, 5)^2$ at position $(3, 5)$ in B .

in B , provided that its area is bigger than or equal to α (steps 6–7). If a match occurs, the largest common square sub-matrix at (i, j) is found (step 8). Otherwise, the square sub-matrix of smaller area at (i, j) is considered to match inside B , provided that its area is bigger than or equal to α (step 10). Increasingly smaller square sub-matrices are verified at (i, j) until an exact match inside B is found or the area of the sub-matrix is smaller than α . If an exact match is not found at (i, j) , the contribution of that position to the similarity measure is zero (steps 13–15). The routine of match $\mathcal{M}(A(i, j)^d, B)$ at step 7 verifies if the square sub-matrix $A(i, j)^d$ has an exact match with some square sub-matrix in B . The area of the largest common square sub-matrices at the different positions in A is progressively summed (step 16). At the end, this value is averaged on the area of A (step 20), and the ACSM similarity measure is returned as output (step 21).

2.3. Example

Figure 1 shows two matrices A and B of size $N = M = 5$ with elements in the alphabet $\Sigma = \{1, 2, 3\}$.

Figure 2 shows the extraction of the largest common square sub-matrix at position $(5, 4)$ in A ($A(5, 4)^2$ of area 2×2 , in red). The maximal square sub-matrix at position $(5, 4)$ is $A(5, 4)^4$ of area $\min\{5, 4\}^2 = 4 \times 4$ (in cyan). The smaller square sub-matrix of area 3×3 at position $(5, 4)$ is $A(5, 4)^3$ (in blue). It is visible that $A(5, 4)^4$ neither $A(5, 4)^3$ exactly match inside B .

$$\begin{array}{ccc}
A(2, 2)^2 = \begin{vmatrix} 1 & 3 \\ 2 & 1 \end{vmatrix} & A(2, 5)^2 = \begin{vmatrix} 1 & 1 \\ 2 & 1 \end{vmatrix} & A(3, 2)^2 = \begin{vmatrix} 2 & 1 \\ 1 & 1 \end{vmatrix} \\
A(3, 3)^2 = \begin{vmatrix} 1 & 3 \\ 1 & 3 \end{vmatrix} & A(3, 4)^2 = \begin{vmatrix} 3 & 2 \\ 3 & 1 \end{vmatrix} & A(3, 5)^2 = \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix} \\
A(4, 2)^2 = \begin{vmatrix} 1 & 1 \\ 2 & 1 \end{vmatrix} & A(4, 4)^2 = \begin{vmatrix} 3 & 1 \\ 2 & 1 \end{vmatrix} & A(4, 5)^3 = \begin{vmatrix} 3 & 2 & 1 \\ 3 & 1 & 2 \\ 2 & 1 & 1 \end{vmatrix} \\
& A(5, 4)^2 = \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix} &
\end{array}$$

Fig. 3. The largest common square sub-matrices between A and B .

Let α be equal to 4. Figure 3 shows the set of the largest square sub-matrices of A exactly matching at some position inside B . Some positions of A are not considered, because of the α value. For example, at position (4, 3) in A the largest square sub-matrix exactly matching inside B is $A(4, 3)^1 = |2|$ of area 1×1 . Because its area is smaller than $\alpha = 4$, it is disregarded.

At the end, the ACSM similarity measure is computed from the area of the largest common square sub-matrices:

$$S_4(A, B) = \frac{W_{\text{tot}}}{N^2} = \frac{4 \times 9 + 9}{25} = 1.8. \quad (2)$$

2.4. Limitations of the ACSM Method and Its Variant A-ACSM

The time complexity of Algorithm 1 is dominated by the exact match procedure $\mathcal{M}(A(i, j)^d, B)$, searching for the exact correspondence of the sub-matrix $A(i, j)^d$ inside B . It can take $O(M^2)$ time, which is independent from the area of the sub-matrix (Crochemore *et al.*, 1995). In the worst case, for each position (i, j) in A (for a total of N^2 positions), a maximum of N sub-matrices can be matched inside B (because of the d parameter). Consequently, the brute-force approach takes $O(M^2 N^3)$ time, which can be further reduced to $O(M^2 N^2 \log_2 N)$ performing a binary search on d (Amelio and Pizzuti, 2013). Although a generalized suffix-tree based solution can be applied on the two matrices, reducing the time complexity to $O(M^2 + N^2)$, the construction of the trie takes $O(M^2(\log_2 M + \log_2 |\Sigma|) + N^2(\log_2 N + \log_2 |\Sigma|))$, which can be quite demanding for large matrices (Amelio and Pizzuti, 2016). Also, the trie data structure is memory-allocated, requiring an extra space of $O(M^2 + N^2)$ (Giancarlo, 1995).

To overcome the current limitations, the A-ACSM similarity measure, an approximate variant of the ACSM similarity measure has been introduced, which omits a portion of elements at regular intervals in the matching process of the sub-matrices (Amelio, 2016). Accordingly, row and column offsets Δ_r and Δ_c determine the size of the interval, which is the distance between the elements to match in the sub-matrices. Although a generalized suffix-tree based solution is not used, the A-ACSM method obtains a lower execution time than Algorithm 1 of ACSM. However, the main limitation of A-ACSM is the size of the

Table 1
 Characteristics and differences of the Common Submatrix approaches. Approximate match indicates that checking the correspondence of some elements is omitted during the match of the sub-matrices.

| Method | Match type | Pattern matching type |
|--------------|--------------------|--|
| ACSM | Exact match | Different positions of the sub-matrices |
| A-ACSM | Approximate match | Different positions of the sub-matrices |
| eACSM | Exact match | Same position of the sub-matrices |

interval, which should be appropriately set in order to avoid information loss in measuring the similarity.

Accordingly, a similarity evaluation which is more efficient than Algorithm 1 and requires any element omission in matching the sub-matrices is still an unsolved problem, and it is the object of our investigation.

3. Extended-Average Common Submatrix Similarity Measure

To overcome the previous limitations, we introduce an optimized version of the Common Submatrix-based similarity measures, which is called *Extended-Average Common Submatrix* (eACSM). Differently from the ACSM measure, it is based on the concept that two matrices are similar to each other if their square sub-matrices match and are located at the same position in the two matrices. This reduces the time complexity of Algorithm 1 of ACSM without using tree indexing data structures. Also, differently from the A-ACSM measure, eACSM exactly matches the sub-matrices because every element of the first sub-matrix is compared with its corresponding element of the second sub-matrix. This avoids to omit the match of some elements inside the sub-matrices, resulting in any information loss in measuring the similarity. Table 1 reports the main differences among the Common Submatrix approaches: (i) ACSM, (ii) A-ACSM, and (iii) eACSM.

This represents a suitable approximation of the similarity when the matrices are images. In other contexts, it may be unavoidable choice when the position of the sub-matrices is associated with specific elements needing to be compared, like various problems in signal processing, image processing, biometry, physics, etc.

3.1. The New eACSM Measure

In Eq. (1), the term $W(i, j)$ is substituted by $\overline{W}(i, j)$, which is the area of the largest square sub-matrix at position (i, j) in A bigger than or equal to α exactly matching a square sub-matrix at the same position (i, j) in B . It is worth noting that $\overline{W}(i, j)$ can be different from $W(i, j)$, because the two largest square sub-matrices at position (i, j) in A matching: (i) at the same position (i, j) in B , and (ii) at some (other) position (h, k) in B , can be different.

Accordingly, the eACSM similarity measure is defined as follows:

$$\bar{S}_\alpha(A, B) = \sum_{i=1}^N \sum_{j=1}^N \frac{\bar{W}(i, j)}{N^2}, \quad \bar{W}(i, j) \geq \alpha. \quad (3)$$

3.2. Modifications in the eACSM Algorithm

The procedure for computing the eACSM similarity measure is shown in Algorithm 2.

Algorithm 2 eACSM similarity measure.

Require: A, B, α

```

1:  $\bar{W}_{\text{tot}} := 0, d := 0, i := 1$ 
2: while  $i \leq N$  do
3:    $j := 1$ 
4:   while  $j \leq N$  do
5:      $\bar{W}(i, j) := 0, d := \min\{i, j\}, \text{flag} := \text{true}$ 
6:     while  $d \geq \sqrt{\alpha}$  and  $\text{flag}$  do
7:       if  $\mathcal{SM}(A(i, j)^d, B(i, j)^d)$  then
8:          $\text{flag} = \text{false}$ 
9:       else
10:         $d = d - 1$ 
11:      end if
12:    end while
13:    if  $\neg \text{flag}$  then
14:       $\bar{W}(i, j) = d^2$ 
15:    end if
16:     $\bar{W}_{\text{tot}} = \bar{W}_{\text{tot}} + \bar{W}(i, j), j = j + 1$ 
17:  end while
18:   $i = i + 1$ 
19: end while
20:  $\bar{S}_\alpha(A, B) = \bar{W}_{\text{tot}}/N^2$ 
21: return  $\bar{S}_\alpha(A, B)$ 

```

The routine $\mathcal{M}(A(i, j)^d, B)$ at step 7 of Algorithm 1 is substituted by the routine of single match $\mathcal{SM}(A(i, j)^d, B(i, j)^d)$. In particular, it computes the exact match between the square sub-matrix $A(i, j)^d$ of area $d \times d$ at position (i, j) in A and the square sub-matrix $B(i, j)^d$ of area $d \times d$ at the same position (i, j) in B . Hence, the match of $A(i, j)^d$ is no more searched at every position inside B . The total area of the largest common square sub-matrices at the same positions in A and B is denoted as \bar{W}_{tot} .

It is worth noting that the eACSM similarity value can be lower than the ACSM similarity value. In fact, matching the sub-matrices in A and B at the same positions is a stricter constraint in the similarity computation than matching at different positions. Nonetheless,

$$A = \begin{vmatrix} 1 & 3 & 2 & 1 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 2 \end{vmatrix} \quad B = \begin{vmatrix} 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 3 & 1 & 2 & 1 \end{vmatrix} \quad A(3,4)^3 = \begin{vmatrix} 3 & 2 & 1 \\ 1 & 3 & 2 \\ 1 & 3 & 1 \end{vmatrix}$$

$$A(3,4)^2 = B(3,4)^2 = \begin{vmatrix} 3 & 2 \\ 3 & 1 \end{vmatrix}$$

Fig. 4. Extraction of the largest common square sub-matrix at position (3, 4) in A . This is $A(3, 4)^2$ of area 2×2 exactly matching the square sub-matrix $B(3, 4)^2$ at the same position (3, 4) in B .

like in the ACSM similarity measure, the eACSM similarity measure ranges in the interval $[0, \sum_{i=1}^N \sum_{j=1}^N \frac{\min\{i, j\}^2}{N^2}]$. This value can be easily normalized dividing by the maximum value.

3.3. Computational Complexity

The substitution of the match routine \mathcal{M} in Algorithm 1 with the single match routine \mathcal{SM} noticeably reduces the time complexity of the procedure.

Theorem 1. *The eACSM algorithm takes $O(N^3)$ time. It can be reduced to $O(N^2 \log_2 N)$ time by performing a binary search on d .*

Proof. Finding the exact match of $A(i, j)^d$ inside B is linear in the area of B , taking $O(M^2)$. Accordingly, checking the exact match of $A(i, j)^d$ at the same position (i, j) in B ($B(i, j)^d$) is constant in the area of B , hence $O(1)$ (Crochemore et al., 1995). This reduces the time complexity from $O(M^2 N^3)$ to $O(N^3)$. Similarly, using the binary search strategy on d (Amelio and Pizzuti, 2013), the time complexity can be further reduced from $O(M^2 N^2 \log_2 N)$ to $O(N^2 \log_2 N)$. \square

Although using a generalized suffix tree takes $O(M^2 + N^2)$ time (Amelio and Pizzuti, 2016), the new method has the advantage to not require the construction and allocation of the tree index. Hence, it is more efficient in this sense.

3.4. Example

Let A and B be the same two matrices depicted in Fig. 1. The largest square sub-matrix at position (3, 4) in A exactly matching at the same position (3, 4) in B is $A(3, 4)^2 = B(3, 4)^2$ of area 2×2 , in red. The sub-matrix of maximal area at position (3, 4) is $A(3, 4)^3$ of area $\min\{3, 4\}^2 = 3 \times 3$ (in blue). It can be observed that $A(3, 4)^3$ does not match at the same position (3, 4) in B .

Figure 5 shows the set of the largest square sub-matrices of A exactly matching at the same positions inside B when α is set to 4. Some positions of A are not considered in the similarity computation, because of the α value. For example, at position (2, 3) in A , the

$$\begin{aligned}
A(2,2)^2 &= \begin{vmatrix} 1 & 3 \\ 2 & 1 \end{vmatrix} & A(3,2)^2 &= \begin{vmatrix} 2 & 1 \\ 1 & 1 \end{vmatrix} & A(3,3)^2 &= \begin{vmatrix} 1 & 3 \\ 1 & 3 \end{vmatrix} \\
A(3,4)^2 &= \begin{vmatrix} 3 & 2 \\ 3 & 1 \end{vmatrix} & A(3,5)^2 &= \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix} & A(4,4)^2 &= \begin{vmatrix} 3 & 1 \\ 2 & 1 \end{vmatrix} \\
A(4,5)^3 &= \begin{vmatrix} 3 & 2 & 1 \\ 3 & 1 & 2 \\ 2 & 1 & 1 \end{vmatrix} & A(5,4)^2 &= \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix}
\end{aligned}$$

Fig. 5. The largest common square sub-matrices at the same positions in A and B .

largest square sub-matrix exactly matching at the same position $(2, 3)$ in B is $A(2, 3)^1 = |3|$ of area 1×1 . Because its area is smaller than $\alpha = 4$, it is discarded.

It can be observed that the sub-matrices $A(2, 5)^2$ and $A(4, 2)^2$ (see Fig. 3) are no more included in the set of the largest common square sub-matrices. This is because $A(2, 5)^2$ and $A(4, 2)^2$ do not match at the same positions $(2, 5)$ and $(4, 2)$ in B , but they match at position $(5, 5)$ in B .

Finally, the eACSM similarity measure can be computed from the area of the largest common square sub-matrices as follows:

$$\bar{S}_4(A, B) = \frac{\bar{W}_{\text{tot}}}{N^2} = \frac{4 \times 7 + 9}{25} = 1.48. \quad (4)$$

4. Experiment

We test the performances of eACSM on two benchmark datasets of handwritten character images. This is accomplished by evaluating the performances of a typical retrieval system where eACSM is employed as the similarity measure. Then, we compare eACSM with other well-known (dis)similarity measures previously introduced in the literature, and show the difference in terms of retrieval performance and execution time.

4.1. Performance Evaluation

Let I_D be an image dataset where the images are categorized in different semantic classes. Also, let Q be a query image not included in I_D and belonging to one of the semantic classes. The aim is to retrieve from I_D the top K most (less) (dis)similar images to Q which are also relevant. Images belonging to the same semantic class of Q are considered relevant. In order to evaluate the retrieval accuracy, we employ the well-known *retrieval precision* measure (Tourassi *et al.*, 2007), which is defined as follows:

$$\text{retrieval precision} = \frac{RI}{K}, \quad (5)$$

where RI is the number of relevant retrieved images and K is the number of retrieved images. In order to compute the retrieval precision, the (dis)similarity between the query

image Q and each image in I_D is computed. Then, the images are sorted in increasing or decreasing order according to the obtained (dis)similarity values. At the end, the relevant images among the top K retrieved images are selected, and their number RI is computed.

Because the retrieval precision is dependent on the selected query image, for a given value of K , we compute this measure for different query images. At the end, we average over the different query images and return the average retrieval precision for the evaluation. Hence, for each value of $K = 1, \dots, N$, where $N \ll |I_D|$, we compute the average retrieval precision for the top K retrieved images.

Also, we evaluate the temporal performances of the (dis)similarity measure by computing the *Average Time per Query (ATQ)*, which is the execution time (in seconds) required for computing the (dis)similarity between a query image Q and each image in I_D averaged over the different query images. Basically, we introduce the *ATQ* as follows:

$$ATQ = \frac{1}{|Q|} \sum_{Q \in Q} T_Q, \quad (6)$$

where Q is the set of considered query images, and T_Q is the total execution time required for computing the (dis)similarity between the current query image Q and all images in I_D .

4.2. Datasets

We perform the evaluation on a randomly selected subset of images from the MNIST dataset.² MNIST is a dataset containing 60'000 training and 10'000 test images of handwritten digits divided into 10 semantic classes (the digits from 0 to 9). The images are extracted from the larger dataset NIST, a benchmark for testing pattern recognition and machine learning methods in image processing and handwritten character recognition. In particular, the training images are composed for one half of some test images from NIST and for the other half of some training images from NIST. The same composition of images is for the test set. Each image is centred and size-normalized.

From the original dataset, we randomly select for the experiment 100 images for each semantic class, in order to obtain a subset of 1000 images with 10 semantic classes. Then, we randomly select 7 query images from the original dataset, not included in the subset of 1000 images and belonging to 4 out of 10 semantic classes. Each image is of area 28×28 in bitmap format (.bmp).

Figure 6 shows sample images from the MNIST dataset.

Also, we perform a second evaluation on a randomly selected subset of images from the NIST Special 19 dataset.³ NIST includes binary images of 3669 handwriting sample forms and 814 255 digit, uppercase and lowercase alphabet letter images segmented from those forms in Portable Network Graphics (.png) format. The area of each segmented image is 128×128 pixels. All segmented images are divided in multiple categories for

²<http://cis.jhu.edu/sachin/digit/digit.html>.

³<https://www.nist.gov/srd/nist-special-database-19>.

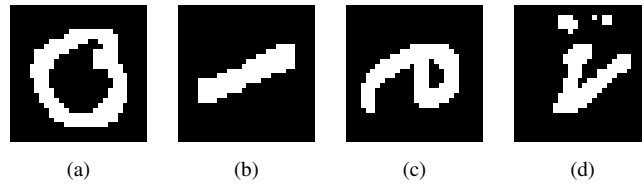


Fig. 6. Sample handwritten digit images from the MNIST dataset: (a) number 0, (b) number 1, (c) number 6, (d) number 7.

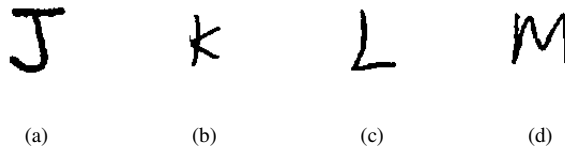


Fig. 7. Sample handwritten alphabet letter images from the NIST Special 19 dataset: (a) letter j, (b) letter k, (c) letter l, (d) letter m.

different recognition tasks: (i) by writer, (ii) by class, (iii) by caseless class (no distinction between uppercase and lowercase letters), and (iv) by field origin.

We extract an image subset corresponding to the only alphabet letters. Basically, we retrieve the relevant images corresponding to the class (letter of the alphabet) of the query image, independently from the writer and the field origin. Accordingly, we use the categorization of the segmented images by class. In this way, the retrieval task is more complex because of the different writing styles and field information.

From the original dataset, we randomly select a subset of 30 segmented images from 7 classes, for a total of 210 images. Also, we randomly extract a set of 5 query images not included in the subset of 210 images, each belonging to one of the 7 classes. Each image is resized to 28×28 pixels and converted in bitmap (.bmp) format.

Figure 7 shows sample images from the NIST Special 19 dataset.

4.3. Competing Methods

Because the Common Submatrix approaches derive from the notion of shared information, which has roots in information theory (Ulitsky *et al.*, 2006), we compare eACSM with other four information-theoretic dissimilarity measures usually adopted in image processing (Tourassi *et al.*, 2007): (i) Joint entropy, (ii) Kullback–Leibler divergence, (iii) Jensen–Shannon divergence, and (iv) Arithmetic-geometric mean divergence.

The Joint entropy (JE) is the entropy of the joint histogram of two images A and B . If the two images are similar, their joint entropy is low in comparison with the sum of their individual entropies. Consequently, JE is considered a distance measure:

$$JE = - \sum_a \sum_b p_{AB}(a, b) \log_2[p_{AB}(a, b)]. \quad (7)$$

The Kullback–Leibler divergence (KL) measures the distance between the probability distribution of one image $a(x)$ and the probability distribution of the second image $b(x)$ (approximated as the image histograms). KL quantifies the average efficiency of adopting the histogram of one image for codifying the histogram of the second one. Consequently, the more similar the two images are, the lower is the KL value:

$$KL(b||a) = \sum_x b(x) \log_2 \frac{b(x)}{a(x)}. \quad (8)$$

The Jensen–Shannon divergence (JD) is a more stable version of KL to noise and variations in histogram binning. Also, it is a symmetric measure and limited between 0 and 2:

$$JD(a, b) = \sum_x \left(b(x) \log \frac{2b(x)}{a(x) + b(x)} + a(x) \log \frac{2a(x)}{a(x) + b(x)} \right). \quad (9)$$

The Arithmetic–Geometric Mean divergence (AGM) quantifies the KL divergence between the arithmetic and geometric means of the probability distributions $a(x)$ and $b(x)$ of the two images:

$$AGM(a, b) = \sum_x \frac{a(x) + b(x)}{2} \log \frac{a(x) + b(x)}{2\sqrt{a(x)b(x)}}. \quad (10)$$

We use the image histograms for approximating the marginal probability distributions of the single images and the joint probability distributions. Accordingly, we set the number of bins of the histograms equal to 256, which is a sufficient number for obtaining a good approximation.

Finally, we compare the eACSM similarity measure with the baseline ACSM similarity measure and its approximate variant A-ACSM. Hence, we set the α parameter of all methods to 5, which revealed the best performances in a similar experiment (Amelio and Pizzuti, 2016). In A-ACSM, we set the parameters of row and column offsets Δ_r and Δ_c respectively to 1 and 4, which experimentally obtained the highest accuracy (Amelio, 2016).

5. Results and Discussion

In the following, we show the results of our experiment on the subset of images from the MNIST and NIST datasets in terms of retrieval precision and execution time. First, eACSM is compared to the baseline ACSM and its approximate variant A-ACSM in terms of retrieval precision. Our aim is to investigate how close the similarity measures are in retrieving relevant images of handwritten characters. Then, eACSM is compared to JE , KL , JD , and AGM in terms of retrieval precision. Finally, eACSM, ACSM and A-ACSM are compared to each other in terms of ATQ in order to understand which similarity measure is the best compromise in execution time and retrieval precision.

Table 2

Retrieval precision obtained by eACSM, ACSM and A-ACSM for the top $K = 10$ most similar images from the subset of images of the MNIST dataset. Δ quantifies the gain of eACSM vs. ACSM and A-ACSM. Δ_{avg} is the average gain of eACSM vs. ACSM and A-ACSM.

| K | eACSM | ACSM | Δ_{ACSM} | A-ACSM | $\Delta_{\text{A-ACSM}}$ |
|-----------------------|-------|-------|------------------------|--------|--------------------------|
| 1 | 1 | 1 | 0 | 1 | 0 |
| 2 | 0.929 | 1 | -0.07 | 1 | -0.07 |
| 3 | 0.905 | 1 | -0.09 | 1 | -0.09 |
| 4 | 0.893 | 0.929 | -0.04 | 0.964 | -0.07 |
| 5 | 0.886 | 0.886 | 0 | 0.971 | -0.08 |
| 6 | 0.881 | 0.905 | -0.02 | 0.952 | -0.07 |
| 7 | 0.878 | 0.898 | -0.02 | 0.939 | -0.06 |
| 8 | 0.875 | 0.911 | -0.04 | 0.929 | -0.05 |
| 9 | 0.873 | 0.889 | -0.02 | 0.921 | -0.04 |
| 10 | 0.857 | 0.886 | -0.03 | 0.914 | -0.06 |
| Δ_{avg} | | | -0.03 | | -0.06 |

5.1. Retrieval Precision Results

5.1.1. MNIST Dataset

Table 2 shows the retrieval precision obtained by eACSM, ACSM and A-ACSM and the gains Δ of eACSM vs. ACSM and A-ACSM for the top $K = 10$ most similar images from the subset of the MNIST dataset. Also, the average gains Δ_{avg} of eACSM vs. ACSM and A-ACSM are reported.

For $K = 1$, eACSM obtains the maximum retrieval precision like ACSM and A-ACSM. For $K = 2, 3, 4$, the retrieval precision of eACSM decreases up to 0.09 vs. ACSM and A-ACSM. Finally, for $K = 5, \dots, 10$, the retrieval precision of eACSM decreases up to 0.04 vs. ACSM and up to 0.08 vs. A-ACSM. On average, the retrieval precision of eACSM only decreases up to 0.03 vs. ACSM and 0.06 vs. A-ACSM. It indicates that matching the sub-matrices that are located at the same position in the two matrices (eACSM) is a good approximation of the baseline matching procedure where the matching sub-matrices can be located at different positions in the two matrices (ACSM and A-ACSM). It is worth noting that the best performances are obtained by A-ACSM, followed by ACSM and eACSM. Also, the Pearson's correlation coefficient computed between eACSM and ACSM is 0.80, indicating a very good correlation between matching the sub-matrices at the same position and at different positions in the two matrices.

Table 3 reports the retrieval precision obtained by eACSM, *JE*, *KL*, *JD*, and *AGM* and the gains Δ of eACSM vs. *JE*, *KL*, *JD*, and *AGM* for the top $K = 10$ most (less) (dis)similar images from the subset of the MNIST dataset. Also, the average gains Δ_{avg} of eACSM vs. *JE*, *KL*, *JD*, and *AGM* are showed.

It is worth noting that eACSM obtains the best performances vs. all competing methods. Basically, it is more competitive with *JE*, with a gain of 0.29 for $K = 1$ and up to 0.14 for $K = 2, \dots, 10$. On average, eACSM gains 0.15 vs. *JE*. On the contrary, eACSM has much better performances than *KL*, *JD* and *AGM*, which have the same retrieval precision over K . Basically, for $K = 1, 2, 3$, eACSM gains up to 1, and for $K = 4, \dots, 10$, eACSM

Table 3

Retrieval precision obtained by eACSM, *JE*, *KL*, *JD*, and *AGM* for the top $K = 10$ most (less) (dis)similar images from the subset of images of the MNIST dataset. Δ quantifies the gain of eACSM vs. *JE*, *KL*, *JD*, and *AGM*. Δ_{avg} is the average gain of eACSM vs. *JE*, *KL*, *JD*, and *AGM*.

| K | eACSM | <i>JE</i> | Δ_{JE} | <i>KL</i> | Δ_{KL} | <i>JD</i> | Δ_{JD} | <i>AGM</i> | Δ_{AGM} |
|-----------------------|-------|-----------|---------------|-----------|---------------|-----------|---------------|------------|----------------|
| 1 | 1 | 0.714 | +0.29 | 0 | +1 | 0 | +1 | 0 | +1 |
| 2 | 0.929 | 0.786 | +0.14 | 0 | +0.93 | 0 | +0.93 | 0 | +0.93 |
| 3 | 0.905 | 0.762 | +0.14 | 0.048 | +0.86 | 0.048 | +0.86 | 0.048 | +0.86 |
| 4 | 0.893 | 0.750 | +0.14 | 0.143 | +0.75 | 0.143 | +0.75 | 0.143 | +0.75 |
| 5 | 0.886 | 0.771 | +0.12 | 0.229 | +0.66 | 0.229 | +0.66 | 0.229 | +0.66 |
| 6 | 0.881 | 0.762 | +0.12 | 0.286 | +0.60 | 0.286 | +0.60 | 0.286 | +0.60 |
| 7 | 0.878 | 0.755 | +0.12 | 0.265 | +0.61 | 0.265 | +0.61 | 0.265 | +0.61 |
| 8 | 0.875 | 0.750 | +0.13 | 0.250 | +0.63 | 0.250 | +0.63 | 0.250 | +0.63 |
| 9 | 0.873 | 0.730 | +0.14 | 0.254 | +0.62 | 0.254 | +0.62 | 0.254 | +0.62 |
| 10 | 0.857 | 0.729 | +0.13 | 0.257 | +0.60 | 0.257 | +0.60 | 0.257 | +0.60 |
| Δ_{avg} | | | +0.15 | | +0.72 | | +0.72 | | +0.72 |

Table 4

Retrieval precision obtained by eACSM, ACSM and A-ACSM for the top $K = 10$ most similar images from the subset of images of the NIST dataset. Δ quantifies the gain of eACSM vs. ACSM and A-ACSM. Δ_{avg} is the average gain of eACSM vs. ACSM and A-ACSM.

| K | eACSM | ACSM | Δ_{ACSM} | A-ACSM | $\Delta_{\text{A-ACSM}}$ |
|-----------------------|-------|-------|------------------------|--------|--------------------------|
| 1 | 0.800 | 0.800 | 0 | 0.600 | +0.20 |
| 2 | 0.900 | 0.800 | +0.10 | 0.600 | +0.30 |
| 3 | 0.933 | 0.867 | +0.07 | 0.533 | +0.40 |
| 4 | 0.850 | 0.850 | 0 | 0.500 | +0.35 |
| 5 | 0.840 | 0.800 | +0.04 | 0.520 | +0.32 |
| 6 | 0.733 | 0.767 | -0.03 | 0.533 | +0.20 |
| 7 | 0.686 | 0.714 | -0.03 | 0.514 | +0.17 |
| 8 | 0.650 | 0.725 | -0.07 | 0.475 | +0.17 |
| 9 | 0.622 | 0.711 | -0.09 | 0.489 | +0.13 |
| 10 | 0.640 | 0.680 | -0.04 | 0.440 | +0.20 |
| Δ_{avg} | | | -0.005 | | +0.24 |

gains up to 0.75 vs. *KL*, *JD* and *AGM*. On average, eACSM gains 0.72 vs. *KL*, *JD* and *AGM*.

5.1.2. NIST Dataset

Table 4 shows the retrieval precision obtained by eACSM, ACSM and A-ACSM and the gains Δ of eACSM vs. ACSM and A-ACSM for the top $K = 10$ most similar images from the subset of the NIST dataset. Also, the average gains Δ_{avg} of eACSM vs. ACSM and A-ACSM are reported.

We can observe that eACSM has better performances than A-ACSM and in some cases than ACSM. Basically, for $K = 1, \dots, 5$, eACSM gains up to 0.10 vs. ACSM. On average, eACSM only decreases up to 0.005 vs. ACSM. It indicates that matching the sub-matrices that are located at the same position in the two matrices (eACSM) is still a good

Table 5

Retrieval precision obtained by eACSM, JE , KL , JD , and AGM for the top $K = 10$ most (less) (dis)similar images from the subset of images of the NIST dataset. Δ quantifies the gain of eACSM vs. JE , KL , JD , and AGM . Δ_{avg} is the average gain of eACSM vs. JE , KL , JD , and AGM .

| K | eACSM | JE | Δ_{JE} | KL | Δ_{KL} | JD | Δ_{JD} | AGM | Δ_{AGM} |
|-----------------------|-------|-------|---------------|-------|---------------|-------|---------------|-------|----------------|
| 1 | 0.800 | 0.800 | 0 | 0.400 | +0.40 | 0.400 | +0.40 | 0.400 | +0.40 |
| 2 | 0.900 | 0.600 | +0.30 | 0.200 | +0.70 | 0.200 | +0.70 | 0.200 | +0.70 |
| 3 | 0.933 | 0.467 | +0.47 | 0.133 | +0.80 | 0.133 | +0.80 | 0.133 | +0.80 |
| 4 | 0.850 | 0.450 | +0.40 | 0.150 | +0.70 | 0.150 | +0.70 | 0.150 | +0.70 |
| 5 | 0.840 | 0.400 | +0.44 | 0.200 | +0.64 | 0.200 | +0.64 | 0.200 | +0.64 |
| 6 | 0.733 | 0.367 | +0.37 | 0.167 | +0.57 | 0.167 | +0.57 | 0.167 | +0.57 |
| 7 | 0.686 | 0.343 | +0.34 | 0.171 | +0.52 | 0.171 | +0.52 | 0.171 | +0.52 |
| 8 | 0.650 | 0.325 | +0.33 | 0.200 | +0.45 | 0.200 | +0.45 | 0.200 | +0.45 |
| 9 | 0.622 | 0.333 | +0.29 | 0.222 | +0.40 | 0.222 | +0.40 | 0.222 | +0.40 |
| 10 | 0.640 | 0.320 | +0.32 | 0.200 | +0.44 | 0.200 | +0.44 | 0.200 | +0.44 |
| Δ_{avg} | | | +0.32 | | +0.56 | | +0.56 | | +0.56 |

approximation of the baseline match at different positions. It is also confirmed by the Pearson's correlation coefficient computed between eACSM and ACSM which is very high and equal to 0.93. In some cases, matching the sub-matrices at the same positions like in eACSM is more effective than the traditional match. In fact, for $K = 1, \dots, 5$, eACSM gains up to 0.40 vs. A-ACSM. Also, for $K = 6, \dots, 10$, eACSM gains up to 0.20 vs. A-ACSM. On average, eACSM gains 0.24 vs. A-ACSM. Hence, exactly matching the sub-matrices that are located at the same position in the two matrices (eACSM) is more effective than approximately matching the sub-matrices that are located at different positions in the two matrices (A-ACSM).

Table 5 shows the retrieval precision obtained by eACSM, JE , KL , JD , and AGM and the gains Δ of eACSM vs. JE , KL , JD , and AGM for the top $K = 10$ most (less) (dis)similar images from the subset of the NIST dataset. Also, the average gains Δ_{avg} of eACSM vs. JE , KL , JD , and AGM are computed.

It is worth noting that eACSM overcomes all competing methods over K . The most competitive method is JE , which obtains the same retrieval precision of eACSM for $K = 1$. Then, for $K = 2, \dots, 10$, eACSM gains up to 0.47 vs. JE . On average, eACSM gains 0.32 vs. JE . On the contrary, KL , JD and AGM have the same retrieval precision over K , which decreases up to 0.80 vs. eACSM. On average, eACSM gains 0.56 vs. KL , JD and AGM .

5.1.3. Summary of the Retrieval Precision Results

Figure 8 illustrates the retrieval precision results reported in the Tables 2–5 for the subset of the MNIST dataset (see Fig. 8 (a)) and for the subset of the NIST dataset (see Fig. 8(b)) and extended for the top $K = 20$ most (less) (dis)similar retrieved images.

We can observe that, for the subset of the MNIST dataset, eACSM, ACSM and A-ACSM have a close trend to each other. For $K = 1, \dots, 10$, this trend is inferiorly limited by eACSM, which is based on the stronger assumption that two matrices are similar if their largest square sub-matrices match at the same position in the two matrices. Hence, it is an

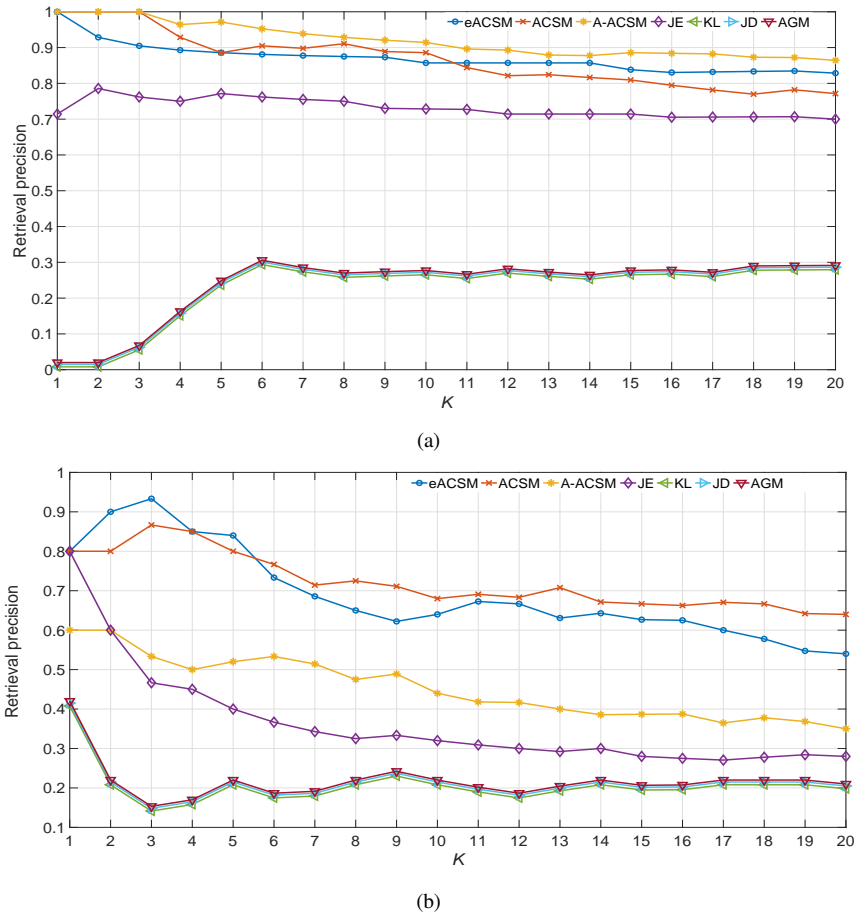


Fig. 8. Retrieval precision obtained by eACSM, ACSM, A-ACSM, JE , KL , JD , and AGM for the top $K = 20$ most (less) (dis)similar images from: (a) the subset of images of the MNIST dataset, and (b) the subset of images of the NIST dataset.

expected result. However, for $K = 11, \dots, 20$, eACSM overcomes ACSM with retrieval precision values above 0.85. As previously observed, the most competing method is JE , with values up to 0.80, while KL , JD and AGM obtain the same lower performances up to 0.3.

For the subset of the NIST dataset, eACSM and ACSM show a very close trend which is never lower than 0.55, while A-ACSM obtains lower performances up to 0.60 which are competitive with JE . On the contrary, KL , JD and AGM have the same performance values up to 0.40 which are much lower than eACSM.

From all aforementioned, the main results obtained by the experiment can be summarized as follows:

1. The eACSM obtains better performances than the other information-theoretic measures in retrieving relevant handwritten character images.

Table 6
 ATQ (in seconds) obtained by eACSM, ACSM and A-ACSM
on the subset of images from the MNIST and NIST datasets.

| Method | MNIST | NIST |
|--------|-------|------|
| eACSM | 3.14 | 0.01 |
| A-ACSM | 20.14 | 2.71 |
| ACSM | 17.28 | 3.57 |

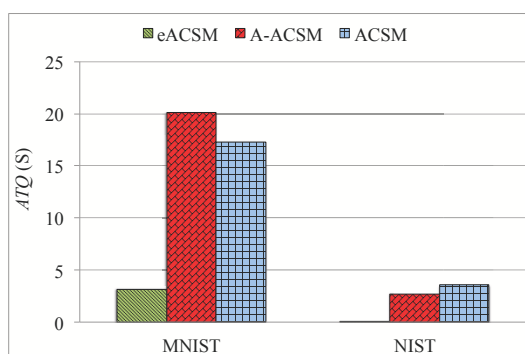


Fig. 9. ATQ graphical results.

2. Matching the sub-matrices at the same position like in eACSM obtains:
 - (a) A precision in retrieving relevant handwritten digit images which is very competitive with the match at different positions like in ACSM and A-ACSM,
 - (b) In some cases, a higher precision in retrieving relevant handwritten letter images than the match at different positions like in ACSM,
 - (c) Higher precision in retrieving relevant handwritten letter images than the match at different positions with omission of some pixels during the match like in A-ACSM.

5.2. Time Performance Results

While the retrieval precision values obtained by eACSM are competitive with those obtained by ACSM and A-ACSM in both the subsets of images from the MNIST and NIST datasets, the temporal performances of eACSM are much better than the other two methods.

In particular, Table 6 reports the ATQ taken by eACSM, ACSM and A-ACSM on the two datasets. Graphical results are depicted in Fig. 9.

It is worth noting that, for the subset of images from the MNIST dataset, the ATQ of eACSM is around 7 times lower than A-ACSM, and around 6 times lower than ACSM. For the subset of images from the NIST dataset, we can observe that the ATQ of eACSM is around 270 times lower than A-ACSM, and around 350 times lower than ACSM. It indicates that eACSM is much more scalable than ACSM and A-ACSM to be employed

for larger datasets. Also, eACSM is the most efficient method to be used for retrieving relevant handwritten character images from a larger dataset.

6. Conclusions

This paper presented eACSM, a new method which is a variant of the Common Submatrix-based similarity measures. It is based on the concept that two matrices are similar to each other if their largest square sub-matrices exactly match at the same position in the two matrices. This noticeably reduced the time complexity of the ACSM brute-force procedure with any element omission in the match of the sub-matrices, like in A-ACSM, neither requiring the construction of a trie, like in ACSM. An experiment performed on a subset of a 1000 images from the MNIST dataset and on a subset of 210 images from the NIST dataset showed that eACSM is very competitive in retrieving handwritten digits and alphabet letters images in terms of retrieval precision and execution time, when compared with other well-known (dis)similarity measures previously introduced in the literature.

Future work will test the proposed method in other contexts where the similarity evaluation is required in the bi-dimensional space, like some problems in physics and biometrics.

Acknowledgements. This work was partially supported by the Grant of the Ministry of Education, Science and Technological Development of the Republic of Serbia, as a part of the project TR33037, and through Mathematical Institute of the Serbian Academy of Sciences and Arts (Project III44006). The baseline ACSM measure was realised at College of Computing, Georgia Institute of Technology (Caianiello Best Young Scientist Paper Award 2013).

References

- Amelio, A. (2016). Approximate matching in ACSM dissimilarity measure. *Procedia Computer Science*, 96, 1479–1488.
- Amelio, A., Pizzuti, C. (2013). Average common submatrix: a new image distance measure. In: *International Conference on Image Analysis and Processing (ICIAP), September 9-13, Naples, Italy, LNCS*, Vol. 8156. Springer, Berlin, Heidelberg, pp. 170–180.
- Amelio, A., Pizzuti, C. (2016). A patch-based measure for image dissimilarity. *Neurocomputing*, 171, 362–378.
- Crochemore, M., Gasieniec, L., Rytter, W., Plandowski, W. (1995). Two-dimensional pattern matching in linear time and small space. In: *Annual Symposium on Theoretical Aspects of Computer Science (STACS), March 2-4, Munich, Germany, LNCS*, Vol. 900. Springer, Berlin, Heidelberg, pp. 181–192.
- Duda, R.O., Hart, P.E., Stork, D.G. (2000). *Pattern Classification*, 2nd edition. Wiley-Interscience.
- Evangelidis, G.D., Psarakis, E.Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10), 1858–1865.
- Fuglede, B., Topsøe, F. (2004). Jensen-Shannon divergence and Hilbert space embedding. In: *International Symposium on Information Theory (ISIT), June 27–July 2, Chicago, IL, USA*. IEEE CS Press, p. 30.
- Giancarlo, R. (1995). A generalization of the suffix tree to square matrices, with applications. *SIAM Journal on Computing*, 24(3), 520–562.
- Goshtasby, A.A., Ardeshtir, A. (2012). *Image Registration: Principles, Tools and Methods*. Springer, London.

- Hill, D.L.G., Hawkes, D.J., Harrison, N.A., Ruff, C.F. (1993). A strategy for automated multimodality image registration incorporating anatomical knowledge and imager characteristics. In: *Biennial International Conference on Information Processing in Medical Imaging (IPMI), June 14–18, Flagstaff, AZ, USA, LNCS*, Vol. 687. Springer, Berlin, Heidelberg, pp. 182–196.
- Kaneko, S., Murase, I., Igarashi, S. (2002). Robust image registration by increment sign correlation. *Pattern Recognition*, 35(10), 2223–2234.
- Kullback, S., Leibler, R.A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.
- Lan, Z.-D., Mohr, R., Remagnino, P. (1995). Robust matching by partial correlation. In: *Proceedings of the 6th British Conference on Machine Vision (Vol. 2) (BMVA), September 11–14, Birmingham, UK*. BMVA Press, pp. 651–660.
- Pearson, K. (1896). Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 187, 253–318.
- Rougon, N.F., Petitjean, C., Preteux, F. (2003). Variational non rigid image registration using exclusive f-information. In: *Proceedings of the International Conference on Image Processing (ICIP), September 14–17, Barcelona, Spain*. IEEE CS Press, pp. 703–706.
- Shannon, C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423.
- Theodoridis, S., Koutroumbas, K. (2009). *Index. Pattern Recognition*, 4th edition. Academic Press, Boston, pp. 949–961.
- Tourassi, G.D., Harrawood, B., Singh, S., Lo, J.Y., Floyd, C.E. (2007). Evaluation of information-theoretic similarity measures for content-based retrieval and detection of masses in mammograms. *Medical Physics*, 34, 140–150.
- Ulitsky, I., Burstein, D., Tuller, T., Chor, B. (2006). The average common substring approach to phylogenomic reconstruction. *Journal of Computational Biology*, 13, 336–350.
- Woods, R.P., Cherry, S.R., Mazziotta, J.C. (1992). Rapid automated algorithm for aligning and reslicing PET images. *Journal of Computer Assisted Tomography*, 16, 620–633.

A. Amelio was awarded the candidate of computer science engineering degree at University of Calabria in 2009 (Faculty of Engineering), doctor of the computer science and systems engineering sciences in 2013. She has served as a contract professor and a research fellow at DIMES University of Calabria. Her research interests have included pattern recognition, artificial intelligence and machine learning algorithms for image processing, document analysis, and social networks.

D. Brodić was awarded the candidate of electrical engineering degree at University of Sarajevo in 1990, doctor of electrical engineering sciences at University of Banja Luka in 2011. He has served as an associate professor at the Technical Faculty in Bor, University of Belgrade. His research interests have included document analysis, artificial intelligence and machine learning algorithms for image processing and pattern recognition.

R. Janković received her BSc and MSc from the Technical Faculty in Bor, University of Belgrade in 2015 and 2016, respectively. Currently, she is a PhD student at the Technical Faculty in Bor, University of Belgrade. Her current research interests include different aspects of statistical analysis and e-business implementation.