Metaclasses and Zoning for Handwritten Document Recognition


Abstract — This work presents a complete method for improving the handwritten document recognition. In this task some characters are confused with others because of their visual/structural similarity. A SOM and TreeSOM neural network were used to sort different characters in metaclasses. In each metaclass a zoning approach was applied trying to get particular features to improve the character classification. The experiments with this new approach were performed in the NIST database with the classic MLP and a fast neural network RBF-DDA.

I. INTRODUCTION

THE document recognition is an important intelligent systems research area. Commercially, several companies use this technology to solve complex real world activities. Character recognition systems are increasingly powerful for printed characters, however much remains to be improved in the handwritten recognition task.

Analyzing the automatic handwritten recognition problem the major issue is the visual/structural similarities among some characters, e.g. the letters “I” and “J”. The use of metaclass is a recognition approach to deals with this type of problem [1]. The metaclass approach builds clusters with similar characters and considers different characteristics using a local strategy to recognize each cluster of characters.

The human strategy to character recognition task is similar to the metaclass approach, we first associates a well known part of the character and then a specialized recognition is used [2]. From this assumption many works have been proposed with a local processing strategy. The use of parts of the characters to extract some local features is called zoning. Some authors proposed empirical zoning [2] [3] and others an automatic zoning [4].

In this paper we propose a new approach to build metaclasses. The metaclasses were created by SOM neural network [6]. The SOM technique allows the creation of clusters containing elements with similar characteristics. To find the best SOM cluster map we used the evaluation technique treeSOM [7]. So the clusters were built according to the best possible cluster composition.

After the metaclasses formation, we have used a zoning mapping approach proposed by Freitas et al. [5] to differentiate characters at each cluster. For each character zoning 118 structural and directional features were extracted.

To verify the performance of the proposed approach we used two classical neural networks classifiers: RBF-DDA [8] and MLP [9].

Different experiments were performed with the NIST database [15]: without using metaclasses and zoning, using metaclasses, using metaclasses and zoning, and using only the zoning approach.

The next section presents the zoning approach. The section 3 details the used feature extraction techniques. Section 4 explains the metaclasses training. The experiments are presented in section 5 and the final remarks are in the last section.

II. THE ZONING MECHANISM

In text recognition the zoning approach is generally defined as the act of divide a standard complex text in several simple parts. So this complex pattern can be recognized by examining these generated simple patterns. With the zoning approach local and combination strategies can be used to simplify the text recognition.

Suen et al. [2] and Li et al. [3] applied the zoning approach to handwritten characters classification. Four different configurations were investigated. In this approach each character is divided in a rectangle with Z parts, where $Z = 2LR$ (L=left, R=right), $2UD$ (U=up, D=down), 4, and 6 zones. In Freitas et al. [5] the zoning approach was investigated using the zones $Z = 4$, $5H$, $5V$ and 7, as presented in the Figure 1.

![Fig. 1. Z = 2LR, 2UD, 4C, 5H, 5V and 7.](image-url)

In this paper we investigate the zoning mechanism (methods of decomposition) by regions. In each zone we
verify the handwritten characters recognition rate. By the analysis of the misclassification in each metaclass it is possible to verify the more relevant zoning models. The zonings with Z = 4, 5H, 5V and 7 in each metaclass generated by the SOM neural network were investigated.

III. METACLASSES CREATION

The metaclasses are classes of classes. This approach has the objective of reduce the recognition task complexity [1]. In handwritten recognition the similar characters are placed in the same cluster so the classification algorithm is trained only with these similar characters. The metaclasses approach has been applied in different character recognition problems. In [1] the authors used metaclasses to recognize dates from bank checks. The months formed by the same suffix, (e.g. November and December) were clustered into the same metaclass.

To built the metaclasses different strategies have been used. Some author used specialist knowledge, metrics, rules, and automatic learning. In [5] the metaclasses are formed by a distance called difference based on the distance (DbD). The DbD distance verifies the difference between the results of two classifiers using confusion matrix. In Table 1 is presented the metaclasses for upper letters found by [5].

We propose the use of a SOM neural network to build the metaclasses. Different SOM topologies were trained with different configurations. The best clusters were found using the strategy proposed by [7].

The Euclidian distance was used to measure the creation of clusters and maps. Two configurations were empirically tested. The maps are: 6x5 and 4x5. Large maps were not used, because our goal is to reduce the number of output classes and large maps make this task more complex.

With the trained SOM network, a stage called discovering the best cluster is performed. The algorithm proposed by Samsonova [7] tries different separation thresholds. With these thresholds different clusters are formed to the same network configuration.

To build the cluster each network node is visited. A new cluster is formed and associated with this node. Successively, their adjacent neighbors are visited.

A distance based on the average values of elements that belongs to each node is calculated between the visited node and their neighbors. If the distance value is less than the threshold, this node is associated with the same cluster of first neighbor node visited. This procedure is performed recursively, until all nodes are visited and associated to the cluster of a neighbor, or a new cluster is created with this node.

The best cluster is calculated after training these clusters for different thresholds. The threshold is randomly chosen between 0 and 1. The chosen values are: 0.63, 0.47, 0.38, 0.25 and 0.18. Each SOM map builds different clusters. This maps many often was formed by different thresholds values of the same SOM map.

The best cluster is the lesser value given by equation 1. The distance between A and B clusters denoted by δAB is equal to the average between pairs of all elements from A and B clusters. The cluster density is denoted by X and means the average among the elements belonging to the same cluster. The number of clusters is denoted by m.

\[
E = m \sum_{A \in E}^{C \in E} \delta_{AB}
\] (1)

The clusters formed by each SOM were created from the observation of the classes that more activates the node of certain clusters. Some clusters are eliminated and others modified. Elements were removed from clusters because the existence of intersections between some clusters with a lot of activation for a given class. Therefore, metaclasses were formed from these clusters.

<table>
<thead>
<tr>
<th>Metaclasses</th>
<th>A,B,C,D,Q,R,S,Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>E,I,J,M,Y</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>G,X</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>F,H,K,L,N,O,P,T</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>U,V,W</td>
</tr>
<tr>
<td>Cluster 5</td>
<td></td>
</tr>
</tbody>
</table>

IV. FEATURE EXTRACTION

One of the most important step in the automatic handwritten character recognition is the features extraction. Such features are used to mapping the relevant characters characteristics into each class.

In literature there are several successful methods to feature extraction in the handwritten recognition task. The most
common feature extraction strategies are the structural and
directional techniques [10][11][12][13]. In this work we
examine the effect of the feature extraction in specific image
regions of Latin alphabet characters (A, B, C, ..., Z).

The selected features are composed by 118 (2 + 76 + 40)
different characteristics composed by Geometric moments,
Concavity measurements and Shape profile respectively.
Each technique is presented in the next subsections.

A. Geometric Moments

The Geometric Moments have been used in pattern
recognition tasks due to its power to extract invariant
features [11][14]. These invariant features are usually related
to character scale, orientation and translation. These
properties are important because the handwritten action and
the image acquisition process may produce image letters
under translation, orientation or scale transformations.

The principal geometric moments are object areas as
center of mass coordinates [14]. We have chosen center of
mass as an important feature for the character separation. We
verified that some misclassified characters have different
center of mass, e.g. C and D.

B. Concavity Measurements

The basic idea in the concavity measurements is to
identify different types of concavities in each character [13].
To verify the concavities from each white pixel in the
character image, we look the 4-freeman directions until
reaches a black pixel or until the image limits. There are four
main concavities elements:

i) non-concavity (no one black pixel was found in the four
directions);

ii) two-directions concavities (two black pixels were found
in two consecutive directions);

iii) three-direction concavities (three black pixels were
found in three consecutive directions);

iv) closed-loops (black pixels were found in the four
directions). More details about this technique can be found
in the Oliveira et al. [13];

C. Shape Representation of Profile

The shape representation of profile is a structural feature
insensitive to the character shape variations [12]. The
character profile shape mainly reflects concavities and
convexities [12]. To calculate the shape profile we use the
object boundary and the corresponding convex hull. The left
and right image boundaries are obtained and their convex
hull are computed. The profile shape is represented as
follows:

\[ DL(y) = Pl(y) - Cl(y) \]  \hspace{1cm} (2)

\[ Dr(y) = Cr(y) - Pr(y) \] \hspace{1cm} (3)

where Pl(y) and Pr(y) are the left and the right profile,
respectively. Cl(y) and Cr(y) are the corresponding convex
hull. Cr(y), Dl(y) and Dr(y) are the structural features.

V. EXPERIMENTAL RESULTS

The experiments were performed with the NIST
handwritten characters database [15]. This is an uppercase
character database composed of 26 class (A-Z). The
database have more than 78,000 images of handwritten
characters. The data distribution is around three thousand
characters per class. All images are at 300 dpi of resolution
with 8 bits per pixel.

In experiments we used 3 subsets: training (50%),
validation (25%) and testing (25%) sets. To evaluate the
metaclases and zoning performance the RBF-DDA and
MLP neural networks were used.

The RBF-DDA network was used because the fast
training. This network does not depend of the network
initialization and does not need a validation set. Also,
preliminary experiments demonstrated that adjustment of \( \theta^+ \)
and \( \theta^- \) parameters, in particular \( \theta^- \), have a great influence
on the network performance [8].

The RBF-DDA network have only one hidden layer and
the training is done in a constructive way. Initially there is
only one hidden layer with no hidden nodes. During the
training new nodes are added to this layer based on the
necessity to represent new data. The used activation function
is the Gaussian function. The classification uses the winner-
takes-all approach.

The Dynamic Decay Adjustment (DDA) training
algorithm performs a small number of epochs, rarely more
than six iterations. As mentioned the parameter \( \theta^- \) directly
influence the network performance. This parameter
represents the activation upper limit between the conflicting
classes. The smaller the \( \theta^- \) value the smaller is the network
error. A lower \( \theta^- \) value produces a greater number of hidden
layer nodes. From a certain limit, when the \( \theta^- \) value is too
small the network tends to memorize the training patterns.
To avoid overfitting we must balance the trade-off between
\( \theta^- \) value and the number of hidden layer nodes. Different
configurations of \( \theta^+ \) and \( \theta^- \) are empirically evaluated and the
best configuration chosen.

The MLP was used because is a classical classifier in
character recognition. The MLP topology was also
empirically defined. In Table 4 is presented the performance
of RBF-DDA and MLP neural networks with and without
metaclases.

These experiments were executed with the metaclases
presented on previous section using the 4x5 and 6x5 SOM
networks. To simplify the table, only the best result are
presented. The best architecture is the SOM 4x5.

The results presented in Table 4 confirm that the use of
metaclases improves the handwritten character
classification. Also, Table 4 shows equivalence between the
RBF and MLP results. The RBF is about 1% better than
MLP. Moreover the RBF was faster than MLP. Based on this
assumption, only the RBF-DDA was used in the next
experiments.
TABLE IV
RESULT OF EXPERIMENTS.

<table>
<thead>
<tr>
<th>Class</th>
<th>Original data</th>
<th>Metaclass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RBF</td>
<td>MLP</td>
</tr>
<tr>
<td>A</td>
<td>92.500%</td>
<td>90.300%</td>
</tr>
<tr>
<td>B</td>
<td>96.491%</td>
<td>92.410%</td>
</tr>
<tr>
<td>C</td>
<td>87.234%</td>
<td>88.920%</td>
</tr>
<tr>
<td>D</td>
<td>90.000%</td>
<td>90.000%</td>
</tr>
<tr>
<td>E</td>
<td>92.308%</td>
<td>91.405%</td>
</tr>
<tr>
<td>F</td>
<td>97.959%</td>
<td>96.536%</td>
</tr>
<tr>
<td>G</td>
<td>100.000%</td>
<td>100.000%</td>
</tr>
<tr>
<td>H</td>
<td>94.828%</td>
<td>93.991%</td>
</tr>
<tr>
<td>I</td>
<td>90.000%</td>
<td>89.561%</td>
</tr>
<tr>
<td>J</td>
<td>41.667%</td>
<td>41.754%</td>
</tr>
<tr>
<td>K</td>
<td>90.909%</td>
<td>90.812%</td>
</tr>
<tr>
<td>L</td>
<td>95.833%</td>
<td>97.513%</td>
</tr>
<tr>
<td>M</td>
<td>83.636%</td>
<td>84.782%</td>
</tr>
<tr>
<td>N</td>
<td>98.077%</td>
<td>98.077%</td>
</tr>
<tr>
<td>O</td>
<td>98.039%</td>
<td>98.039%</td>
</tr>
<tr>
<td>P</td>
<td>96.000%</td>
<td>96.450%</td>
</tr>
<tr>
<td>Q</td>
<td>89.286%</td>
<td>88.286%</td>
</tr>
<tr>
<td>R</td>
<td>92.727%</td>
<td>92.727%</td>
</tr>
<tr>
<td>S</td>
<td>92.157%</td>
<td>92.157%</td>
</tr>
<tr>
<td>T</td>
<td>92.308%</td>
<td>92.308%</td>
</tr>
<tr>
<td>U</td>
<td>85.714%</td>
<td>85.714%</td>
</tr>
<tr>
<td>V</td>
<td>95.918%</td>
<td>95.918%</td>
</tr>
<tr>
<td>W</td>
<td>79.487%</td>
<td>77.249%</td>
</tr>
<tr>
<td>X</td>
<td>97.917%</td>
<td>97.920%</td>
</tr>
<tr>
<td>Y</td>
<td>92.500%</td>
<td>92.500%</td>
</tr>
<tr>
<td>Z</td>
<td>97.917%</td>
<td>97.917%</td>
</tr>
</tbody>
</table>

Mean 87.058% 90.509% 97.429% 97.161%

Desv. 0.209 11.158 0.028 2.940

This experiment presented interesting results. Note that some letters can be classified better in differentzonings approaches. For 4C zoning the best results was obtained in the letters A, B, C, D, I, J, K, S, V, Y and Z. The letter D obtained the worst result 92.593% of success recognition. In the major of the letters the precision rate was about 100%. The average of all letters in this zoning was the best zoning result (96.201% of true recognition). In 5V zoning the letters B, C, E, F, G, H, N, O, R, U, X and Z. The average rate of all letters of this zoning approach was close the best 4C result. The 5H zoning provided the best results for C, G, L, M, Q, T and U. Considering all the letters, the 5H zoning produced the worst result. The 7 zoning approach produce best results for B, C, G, H, P, Q, R, V, W and X letters (see Table 5).

The results presented in Table 5 suggested that the metaclasses method is hampering the zoning performance. To verify this hypothesis an experiment using only the zoning approach was performed.

The Table 6 presents the results of the zoning experiments. Using the same features and the RBF-DDA network the obtained results was worse than the results using metaclasses and zoning. Because this we choose to show only the mean and standard deviation for each zoning approach.

TABLE VI
ZONING WITHOUT METACLASSES.

<table>
<thead>
<tr>
<th>Z = 4C</th>
<th>Z = 5V</th>
<th>Z = 5H</th>
<th>Z = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>96.078%</td>
<td>94.000%</td>
<td>90.000%</td>
</tr>
<tr>
<td>Desv.</td>
<td>100.000%</td>
<td>100.000%</td>
<td>100.000%</td>
</tr>
</tbody>
</table>

To refine the experiments, the clusters were compared separately. The objective is to understand the behavior of each method, zoning and metaclasses, in the cluster classification. In Table 7 this comparison is presented. It can be see that the best result belongs of the metaclasses approach. Only for cluster 3 the zoning approach is better. Comparing only the zonings approaches, the 5V was the best one. The 5V zoning is better in the cluster 3. Following this thought, the 4C is the best one for the cluster 1. For the cluster 2 the best zoning is 5H. For the clusters 4 and 5 the best zonings approaches are 5V and 4C, respectively.

TABLE VII
RESULT OF EXPERIMENTS WITH THE DDA-RBF.

<table>
<thead>
<tr>
<th>Clus.</th>
<th>Metaclasses</th>
<th>Z = 4C</th>
<th>Z = 5V</th>
<th>Z = 5H</th>
<th>Z = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>97.98%</td>
<td>96.78%</td>
<td>97.15%</td>
<td>91.66%</td>
<td>96.12%</td>
</tr>
<tr>
<td>Desv.</td>
<td>0.016</td>
<td>0.020</td>
<td>0.025</td>
<td>0.068</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Considering a combination approach the use of the 5V with the metaclass 3 needs to be chosen. Consequently the best obtained result 97.983% for the metaclasses method, grow to 98.14% using the metaclasses zoning combination.
VI. Final Remarks

In this paper we presented a new method to build metaclasses. We have used the TreesSOM approach for choosing the best map configuration created by the SOM neural network to create metaclasses to classify handwritten characters. Different zonings approaches were combined with the metaclasses strategy trying to improve the recognition precision. We used a specialized feature extraction in the area of each zoning division. The neural networks MLP and RBF-DDA were used as classifiers.

With the metaclasses method, in all classes we obtained an improvement in the classification about more than 10% of precision. The performance of the classifiers using the zoning approaches was very close to the performance using only metaclasses. Therefore, we can conclude the zoning approach does not improve the metaclasses performance.

The use of the zonings approaches alone does not presented better results, so the metaclasses help to improve the efficiency of this approach. If it is necessary the use of the zoning approach we can conclude that the combination with metaclass approach will improve the classifier precision.

Finally we performed a comparison with each cluster generated by metaclasses and verified the zoning approach only is better in the cluster 3 of the metaclasses. So the use of zoning approach is interesting only to letters of the cluster 3.

Consequently, we can say the metaclasses approach presented in this paper helps to improve the handwritten recognition. The effort to make zoning approach is not justified by the obtained results. Furthermore, zoning is a slow approach, since we have more features from each zoning area to extract and the achieved feature vector is bigger and then the training is slower.

The most important advantage of the metaclasses approach is the considerable improvement of the recognition rate with a simple method of clusters evaluations.

As future work we consider to verify the method performance on lowercase letters and investigate the use of others cluster evaluation methods.

ACKNOWLEDGMENT

The authors would like to thank FACEPE (Brazilian research agencies) for their financial support.

REFERENCES