Inventory-Extraction-Tool Prototype v. 0.2

User Manual

Sebastian Colutto

sebastian.colutto@uibk.ac.at

Table of contents

1. Preface

The Inventory-Extraction-Tool is a prototype software that is developed during the EU 7th Framework Program grant IMPACT (Improving Access to Text). Its intended use is to extract an inventory of the characters out of a set of input documents. The inventory essentially is a clustering of the characters into groups of similar looking symbols. The clustering is computed by comparing symbols according to their visual appearance. The main objective of the tool is to rapidly create a training set of characters to digitize documents with unknown font type. It therefore also includes a post-correction and labeling module, where inventories computed by performing cluster analysis on the set of input glyphs can be post-corrected and labeled in order to create a training set for further classification of documents with the same font type.

In this user manual, the most prominent features of the Inventory-Extraction-Tool v.0.2 are explained and outlined with screenshots.

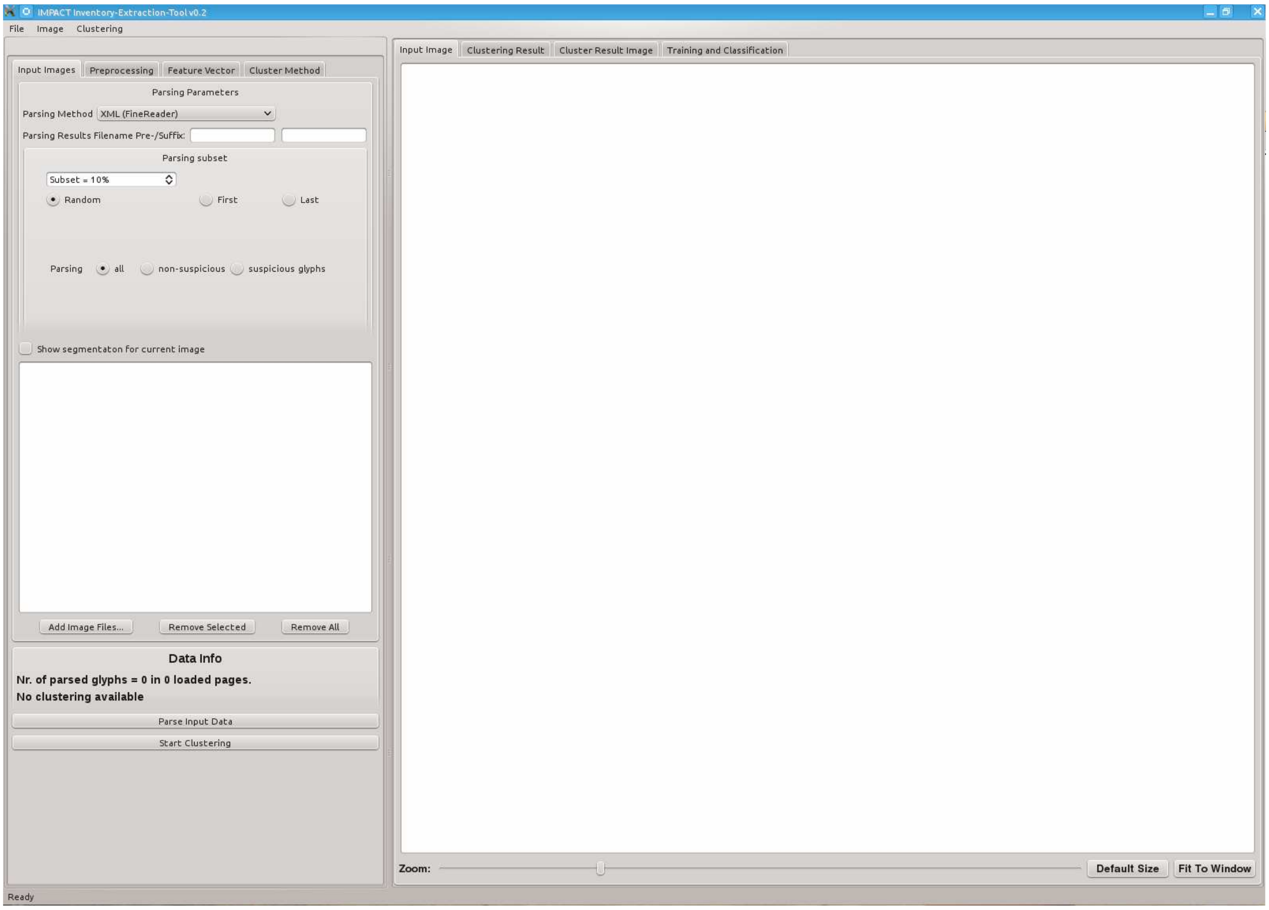
We also want to note, that the tool has prototype status only, which in particular means that there is no guarantee for stability. Furthermore, the tool is work in progress and thus changes in the functionality and / or graphical user interface can occur at any time.

2. Copyright and License

3. Main application window

A screenshot of the main application window can be seen in Figure 1. The graphical user interface is divided into two main areas: the left tab window area where input files can be loaded and the parameters for the clustering algorithms and feature vector computation can be set and a right tab window area where input files and clustering results are visualized. Additionally, in the right tab window area, clustering results can be edited and labeled after the clustering process.

The window bar at the top includes additional functionalities, most prominently loading and saving of clustering results to and from XML files.



Parsing Parameters

Parsing Method: XML (FineReader)

Parsing Results Filename Pre-/Suffix:

Parsing subset

Subset = 10%

● Random ○ First ○ Last

Parsing ● all ○ non-suspicious ○ suspicious glyphs

□ Show segmentaton for current image

Data Info

Nr. of parsed glyphs = 0 in 0 loaded pages.

No clustering available

Ready

Figure 1: Screenshot of the main application window

4. Loading of input images

To inventory extraction tool needs preprocessed and binarized images and a segmentation on character level as input.

To load input files into the tool and set the parameters for parsing the characters, switch to the tab '*Input Images*' on the left tab window area, as can be seen in Figure 2.

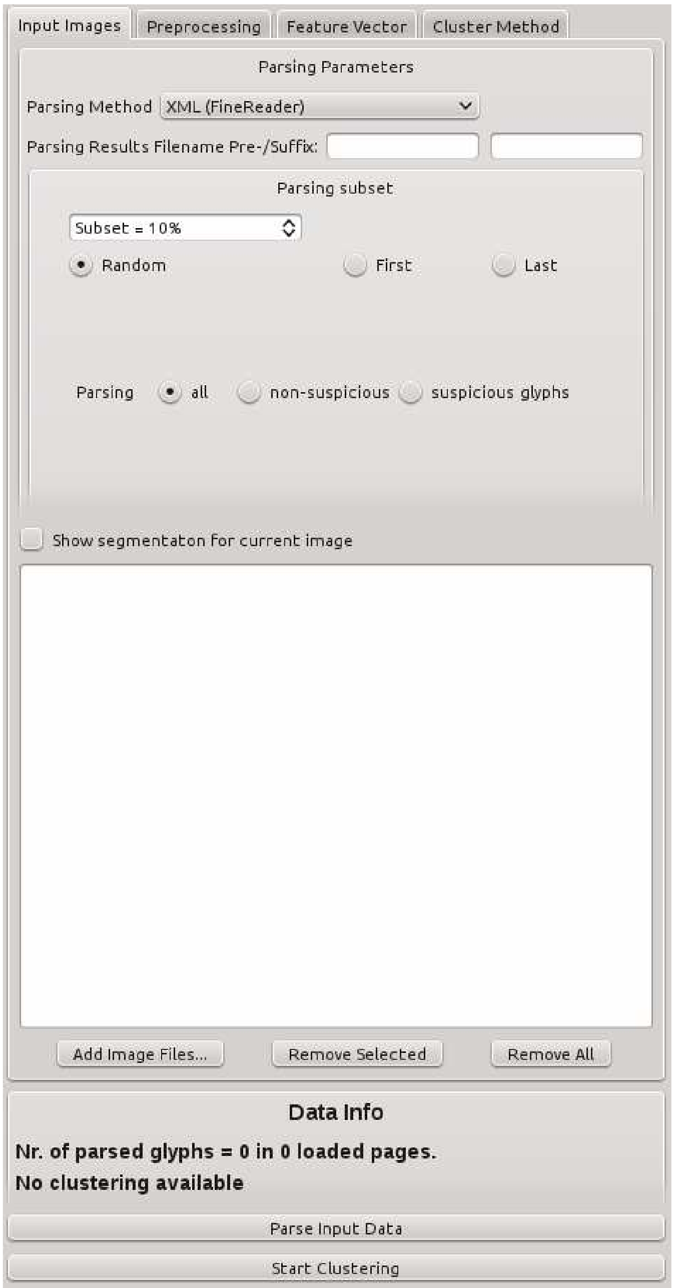
Click on the button '*Add Image Files...*' to add images to the list of input images. Those are the input files, where the characters are parsed from. The segmentation result files for the input images have to be in the same directory as the images and their filename (without file extension) has to be the same as the name of the image. Additionally, one can specify segmentation result filename pre- or suffixes in the GUI if needed. To view one of the input images, double-click on the corresponding entry in the list and it will be displayed on the right tab window area in the tab '*Input Image*'. By checking the '*Show segmentation for current image*' box, one can additionally display the segmentation results for the currently loaded image. The segmentation results are visualized using rectangular bounding boxes using a red color for non-suspicious glyphs and a blue color for suspicious glyphs.

The segmentation result type can be set by selecting the corresponding type in the 'Parsing Method' box. Currently three segmentation results types are supported: XML files created by FineReader and the Im2CharRects tool of IBM and DAT files created from the segmentation tools created by NCSR within the IMPACT project.

Using the box '*Parsing subset*' one can determine, which subset of the characters are actually parsed and loaded into the tool. One can choose how much percent of the characters per page are parsed and if a random selection, the first or the last characters of a page are parsed. Furthermore, we are able to choose, if we want to parse all characters, only suspicious or non-suspicious glyphs. Note, that this is only reasonable for XML segmentation result files from FineReader, where the suspicious flag is available.

After all parameters are set, click on the button '*Parse Input Data*' and the parsing process will be begin. After it is finished, the number of parsed glyphs and the number of pages is displayed in the information area on the bottom of the left tab window area.

Figures 2-4 show different stages of the input parsing process.



Parsing Method: XML (FineReader)

Parsing Results Filename Pre-/Suffix:

Subset = 10%

● Random ○ First ○ Last

Parsing ● all ○ non-suspicious ○ suspicious glyphs

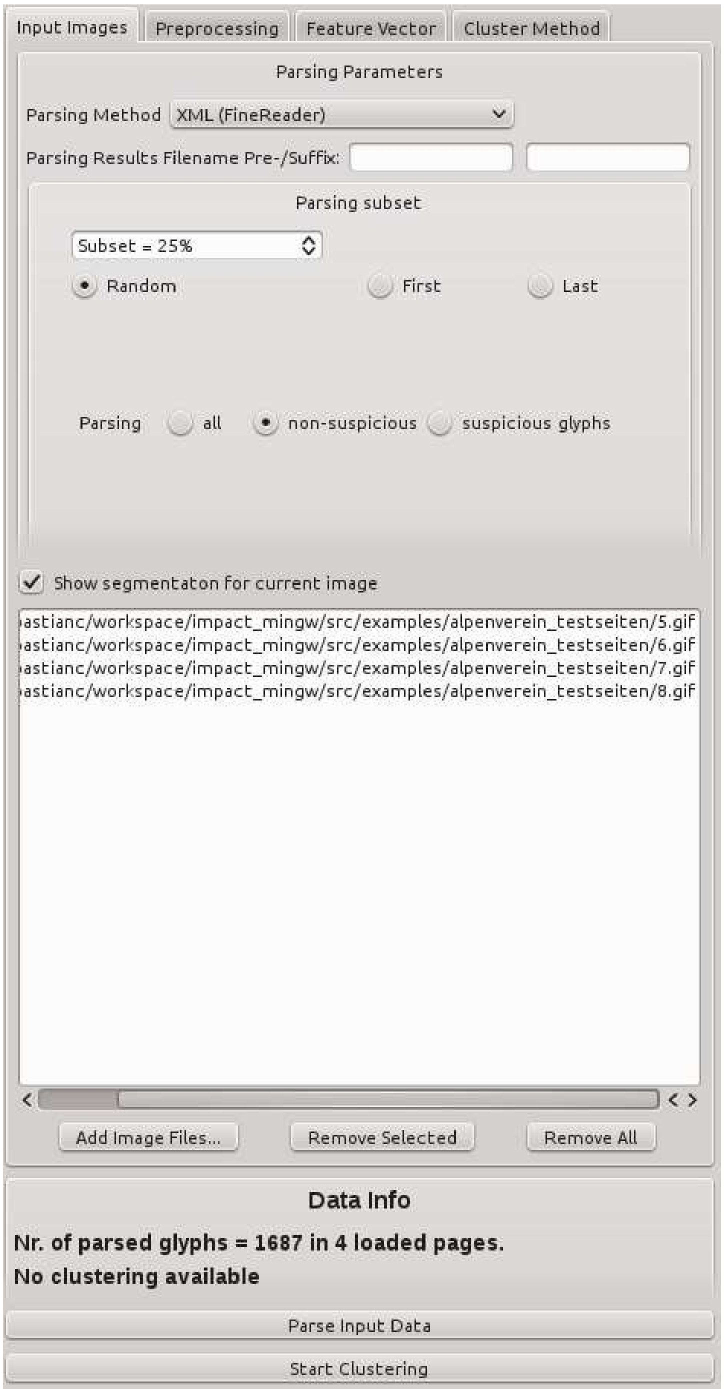
□ Show segmentaton for current image

Data Info

Nr. of parsed glyphs = 0 in 0 loaded pages.

No clustering available

Figure 2: Initial input images area



Parsing Method: XML (FineReader)

Parsing Results Filename Pre-/Suffix:

Subset = 25%

● Random ○ First ○ Last

Parsing ○ all ● non-suspicious ○ suspicious glyphs

☑ Show segmentaton for current image

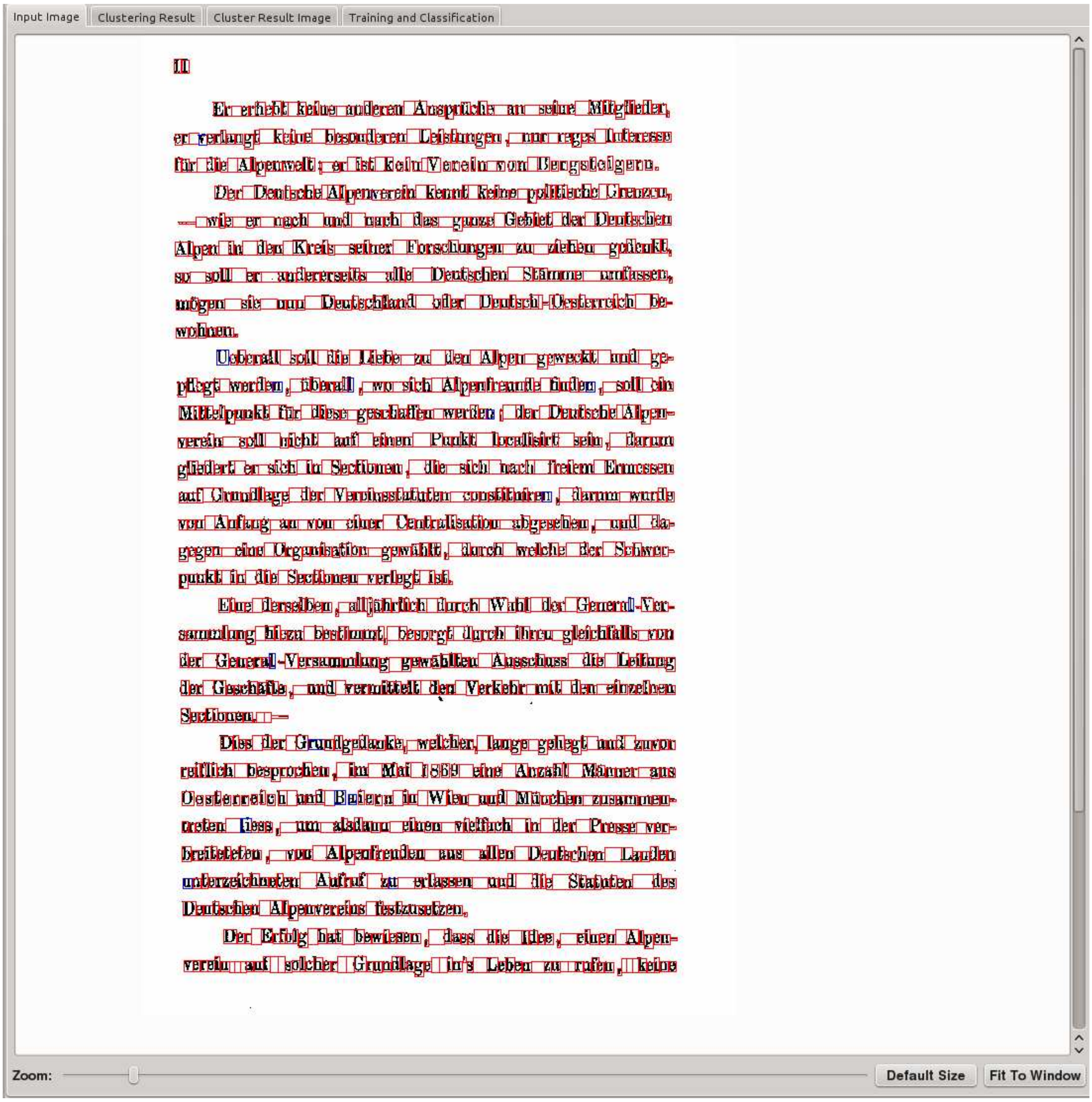
```
astianc/workspace/impact_mingw/src/examples/alpenverein_testseiten/5.gif
astianc/workspace/impact_mingw/src/examples/alpenverein_testseiten/6.gif
astianc/workspace/impact_mingw/src/examples/alpenverein_testseiten/7.gif
astianc/workspace/impact_mingw/src/examples/alpenverein_testseiten/8.gif
```

Data Info

Nr. of parsed glyphs = 1687 in 4 loaded pages.

No clustering available

Figure 3: Input images area with some input files and parsed characters. 1687 glyphs were parsed out of 4 images, which was a random selection of 25% of characters from each page of the input images. Only non-suspicious glyphs were parsed out of the Fine



II

Er erhebt keine anderen Ansprüche an seine Mitglieder, er verlangt keine besonderen Leistungen, nur reges Interesse für die Alpenwelt; er ist kein Verein von Bergsteigern.

Der Deutsche Alpenverein kennt keine politische Grenzen, — wie er nach und nach das ganze Gebiet der Deutschen Alpen in den Kreis seiner Forschungen zu ziehen gedenkt, so soll er andererseits alle Deutschen Stämme umfassen, mögen sie nun Deutschland oder Deutsch-Oesterreich bewohnen.

Ueberall soll die Liebe zu den Alpen geweckt und gepflegt werden, überall, wo sich Alpenfreunde finden, soll ein Mittelpunkt für diese geschaffen werden; der Deutsche Alpenverein soll nicht auf einen Punkt localisirt sein, darum gliedert er sich in Sectionen, die sich nach freiem Ermessen auf Grundlage der Vereinsstatuten constituiren, darum wurde von Anfang an von einer Centralisation abgesehen, und dagegen eine Organisation gewählt, durch welche der Schwerpunkt in die Sectionen verlegt ist.

Eine derselben, alljährlich durch Wahl der General-Versammlung hiezu bestimmt, besorgt durch ihren gleichfalls von der General-Versammlung gewählten Ausschuss die Leitung der Geschäfte, und vermittelt den Verkehr mit den einzelnen Sectionen. —

Dies der Grundgedanke, welcher, lange gehegt und zuvor reiflich besprochen, im Mai 1869 eine Anzahl Männer aus Oesterreich und Baiern in Wien und München zusammentreten liess, um alsdann einen vielfach in der Presse verbreiteten, von Alpenfreunden aus allen Deutschen Landen unterzeichneten Aufruf zu erlassen und die Statuten des Deutschen Alpenvereins festzusetzen.

Der Erfolg hat bewiesen, dass die Idee, einen Alpenverein auf solcher Grundlage in's Leben zu rufen, keine

Figure 4: Visualized input image including segmentation results

5. Preprocessing parameters

In order to compare characters by their visual appearance with high accuracy, all glyph images have to be preprocessed, which is done before the actual clustering. Parameters for the preprocessing can be set in the 'Preprocessing' tab of the left tab window area (see Figure 5).

Default values for the preprocessing are set and need not to be changed in most cases.

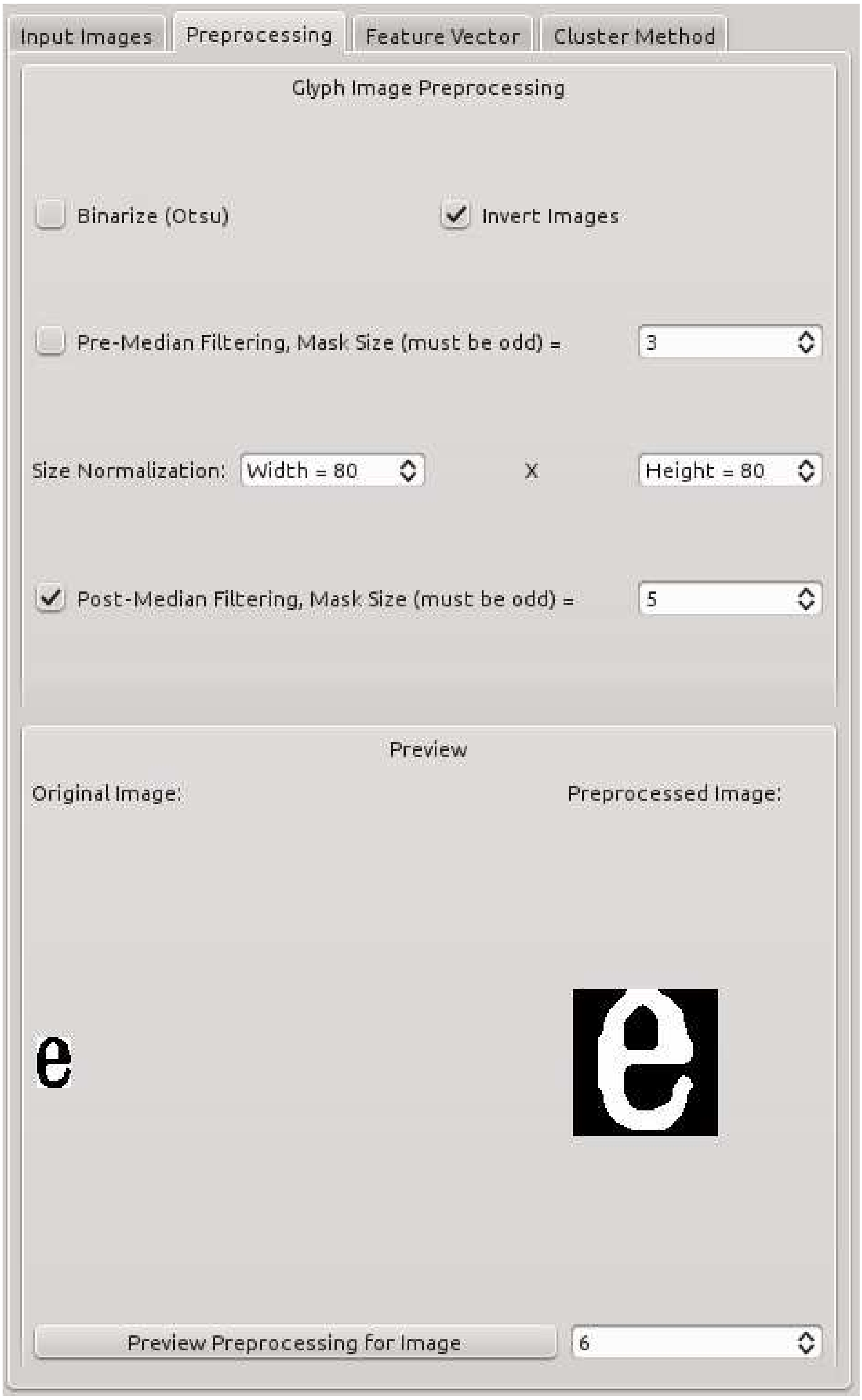
Using the '*Binarize*' option, the user can select, if the input images are binarized using the Otsu-method, in case the input images were not binarized already. This is not recommended however, since a more sophisticated binarization method is needed for more complex documents.

Input glyphs are then inverted by default (using the '*Invert Images*' checkbox), such that white pixels are foreground pixels and black pixels are background.

By checking the '*Pre-Median Filtering*' or '*Post-Median Filtering*' checkboxes, the user can choose, if the glyphs are Median filtered before or/and after the size normalization process with corresponding mask sizes. Such a filtering can be useful, if input glyphs with a high noise rate are given.

The core part of the preprocessing is the size-normalization process. Input glyphs are size normalized to a size specified in the '*Size Normalization*' box (80 x 80 pixels by default), such that the center of the image corresponds to the center of mass of the character. Size normalization is done using a bi-cubic interpolation process.

To preview the results of all preprocessing steps for the currently set parameters for one character, one can click on the '*Preview Preprocessing for Image*' button in the '*Preview*' area on the bottom of the tab and specify the number of the parsed character to preview the preprocessing for this glyph. The original parsed character image is shown on the left, while the right image shows the preprocessing result.



Glyph Image Preprocessing

□ Binarize (Otsu) ☑ Invert Images

□ Pre-Median Filtering, Mask Size (must be odd) = 3

Size Normalization: Width = 80 X Height = 80

☑ Post-Median Filtering, Mask Size (must be odd) = 5

Preview

Original Image: Preprocessed Image:

Preview Preprocessing for Image 6

Figure 5: Preprocessing parameters area. A preprocessing result for the character number 6 (i.e. the sixth character that was parsed) is shown in the preview area. Note the smooth result of the preprocessed glyph image due to the post median filter using mask size 5.

6. Feature vector computation

In addition to the default distance computation between input glyphs which is performed using a pixel-wise dissimilarity measure that has been developed for this purpose, the tool also provides the computation of standard feature vectors. Those feature vectors are calculated for every preprocessed input glyph and can then by used to compute the distance between two glyphs using the Euclidean distance between the vectors corresponding to the characters. Furthermore, those feature vectors will be used in a future version of the tool that aims to include training and subsequent classification of documents with the training set, created by the tool.

Note that for a default clustering workflow, feature vectors need not to be specified and this tab can therefore be omitted.

To set the parameters for the feature vectors, switch to the '*Feature Vector*' tab on the left tab window area. A screenshot of the widget is shown in Figure 6. On the top of the tab, the different features that are supported are listed group-wise. The currently supported features are basic-image moments, AMI moments (affine invariant image moments), Hu image moments and profile features. For the profile features, one can specify the number of blocks that are created in the profile and if inner or outer profiles are created in a horizontal or vertical fashion. To add specific features to the feature vector, push the corresponding '*Add to Feature Vector*' button and the feature, including information about its dimensionality is added to the list in the middle of the widget, where certain features can also be removed again. All of the features in the list then compose the final feature vector.

Additionally, one can choose to perform dimensionality reduction on the feature vectors using principal component analysis by checking the '*Dimension Reduction*' box. There, a threshold on the variability of the original data that the resulting vectors should cover can be set or the dimension of the final feature vectors can be specified by hand.

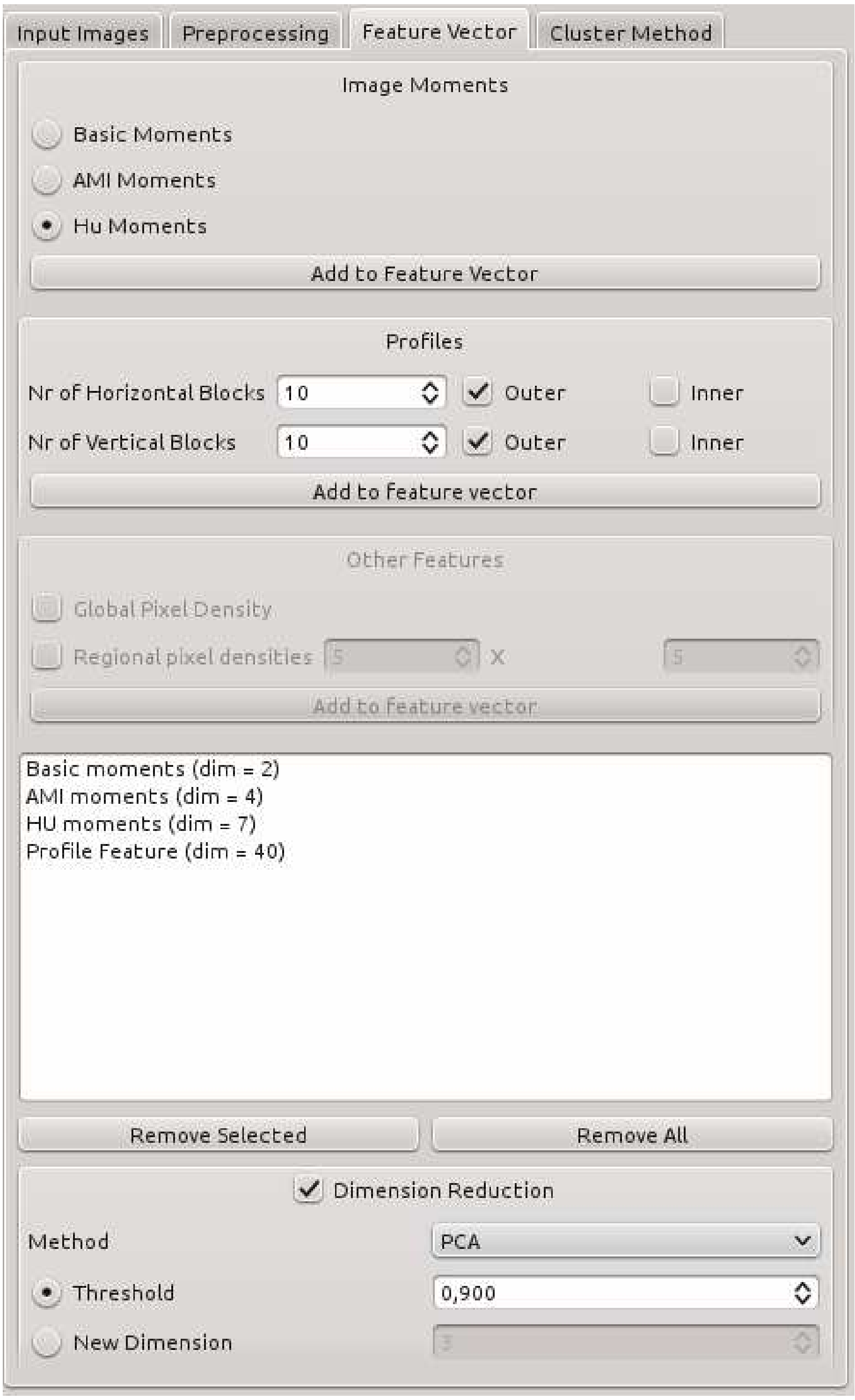


Image Moments

○ Basic Moments
○ AMI Moments
● Hu Moments

Profiles

Nr of Horizontal Blocks 10 ☑ Outer □ Inner
Nr of Vertical Blocks 10 ☑ Outer □ Inner

Other Features

□ Global Pixel Density
□ Regional pixel densities 5 X 5

Basic moments (dim = 2)
AMI moments (dim = 4)
HU moments (dim = 7)
Profile Feature (dim = 40)

☑ Dimension Reduction

Method PCA

● Threshold 0,900
○ New Dimension 3

Figure 6: Feature vector tab. A feature vector including basic image moments, AMI moments, HU moments and Profile Features will be created. Additionally, dimensionality reduction will be performed on the feature vectors, such that the resulting vectors cover 90% of the variability of the original data.

7. Clustering parameters

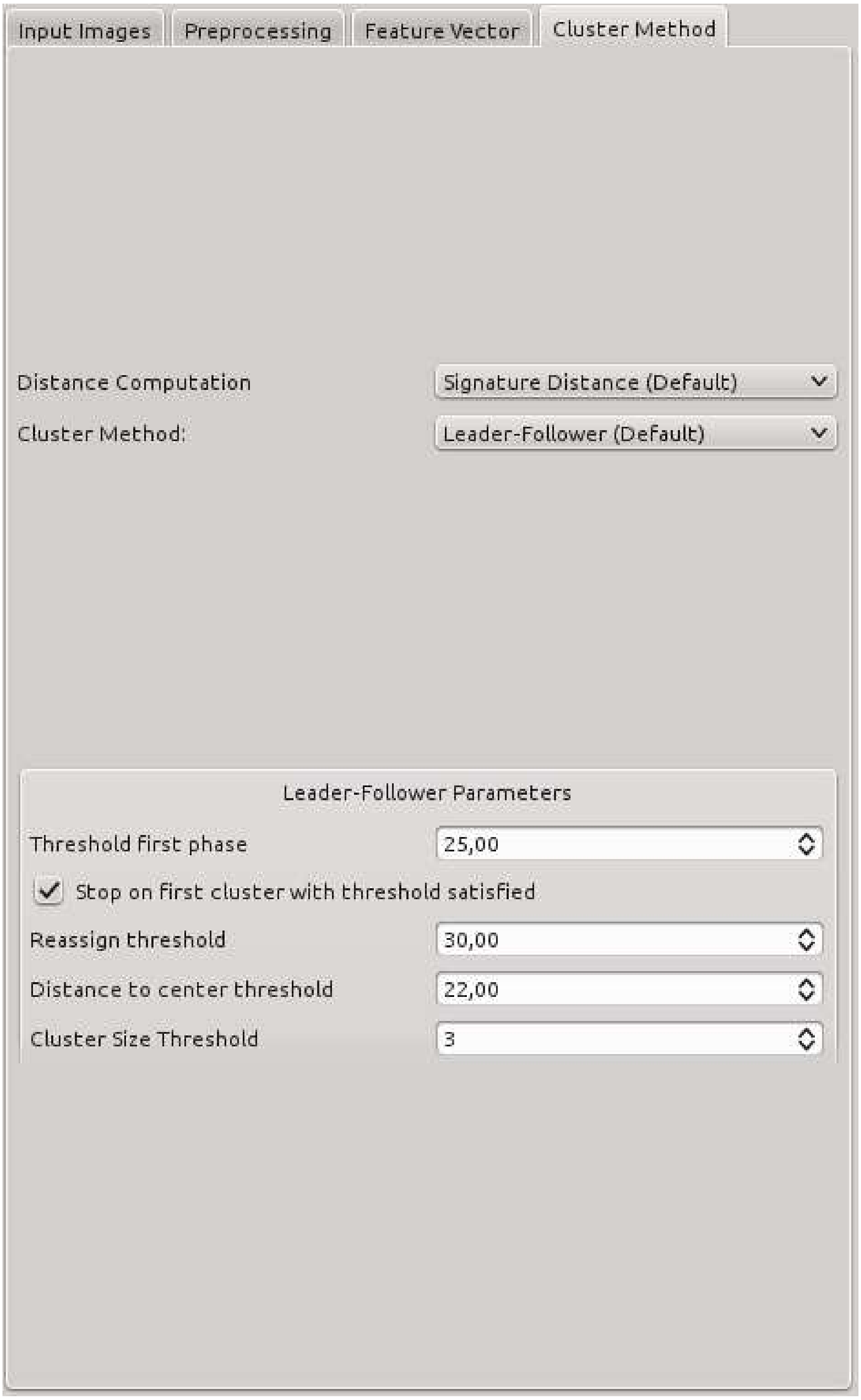
To set the parameters for the clustering process, switch to the tab '*Cluster Method*' on the left tab window area. A screenshot of the '*Cluster Method*' tab is shown in Figure 7.

On the first row of the top area, one can choose the type of distance computation by changing the value of the '*Distance Computation*' box. By default, the distance is computed using the '*Signature Distance*' which is the special dissimilarity measure that was developed for this purpose. On the other hand, clustering can also be performed using feature vectors. To this end, one has to create a feature vector using the '*Feature Vector*' tab on left tab window area which was explained in more detail in the last chapter.

Once the distance computation type has been set, one can choose the cluster method by using the box in the '*Cluster Method*' row. Currently a modified version of the Leader-Follower clustering algorithm (default), agglomerative hierarchical clustering, DBSCAN and K-Means are supported. K-Means clustering however is only supported when using feature vectors for distance computation. Note that all clustering algorithms except the default Leader-Follower algorithm using the Signature-Distance are for experimental purpose only. We therefore only describe the parameters of this clustering algorithm.

The parameters of the Leader-Follower clustering algorithm can be set on the bottom of the clustering parameters tab. The first parameter ('*Threshold first phase*') is the threshold that is used for a first phase of Leader-Follower clustering which extracts a set of prototype characters from the input set. Additionally one can choose, if the assignment process of the Leader-Follower clustering stops for each character on the first cluster with the specified threshold satisfied or if all current cluster prototypes are compared and the one with the smallest distance is chosen ('*Stop on first cluster with threshold satisfied*' box). The second threshold parameter ('*Reassign threshold*') determines which characters are reassigned to a new cluster or thrown away. Each character having a distance greater than this threshold to any of the current prototypes is thrown into a junk cluster, which is the first cluster in the clustering result. The third threshold parameter ('*Distance to center threshold*') determines, which characters are chosen for reassignment. Only characters with a distance larger than this threshold are reassigned in the second step of the clustering process. Finally, the last parameter ('*Cluster Size Threshold*') determines, which clusters are throw away after the second clustering stage. All clusters with size less or equal to this parameter are deleted and its characters are added to the junk cluster.

Figure 7: Cluster parameters widget. Leader-Follower clustering using the Signature-Distance is used for clustering.



Distance Computation: Signature Distance (Default)

Cluster Method: Leader-Follower (Default)

Leader-Follower Parameters

Threshold first phase: 25,00

☑ Stop on first cluster with threshold satisfied

Reassign threshold: 30,00

Distance to center threshold: 22,00

Cluster Size Threshold: 3

8. Visualization and editing of clustering results

Once all parameters are set, the user can click on the '*Start clustering*' button in the bottom left of the graphical user interface to start the clustering process with the parsed characters and all the parameters for preprocessing, feature vector computation and clustering. After the clustering is finished, the result will be visualized in the '*Clustering Result*' tab on the right tab window area (see Figure 8).

The '*Clustering Result*' tab is divided into two main parts:

- **Prototype view area**: In the top area, all clusters are visualized using their prototypes, which are averaged images of the first 20 members of the clusters. Additionally, in this view, the cluster labels (a four digits hexadecimal numbers) are displayed below the cluster prototype image. This view enables the user to quickly browse through the created clusters and detect severe cluster errors immediately by looking at their corresponding prototypes in the top area. One can also use drag-and-drop operations on this window to quickly merge clusters with the same characters, e.g. two different clusters with 'e' letters. Multiple cluster selections are also supported here using the default shortcut combinations.
- **Cluster view area:** In the bottom area, the content of one cluster is displayed. The cluster content for a specific cluster can be displayed by double-clicking on the corresponding prototype symbol of the cluster in the top view of this widget. By clicking on one of the characters in the list of the cluster content window, information about it is shown in the bottom left area of this widget, which contains the character bounding box, its internal parsing ID, the corresponding image ID, the recognized text (if available) and its distance to the corresponding cluster center. Editing the content of the visualized cluster is also possible using drag-and-drop operations. Characters inside the cluster can be selected and added to another cluster by dragging them to the corresponding cluster of the prototype view are. A visualized cluster can also be removed from the clustering result by clicking on the '*Remove*' button on the bottom of this widget. Additionally, one can click on the button ‘Move selected elements to trash’ to tag the selected elements as noise and move them to the trash cluster.

Additional functionality for visualization and editing of the clustering can be accessed by the buttons in the mid-area of the window. They are grouped into '*Editing options*' and '*Viewing options*' corresponding to functions that edit the clustering result or change the view of the current clustering in the prototype view area.

The buttons of the editing options are:

- '*Label!*' - This button labels the currently displayed cluster with the four digits hexadecimal number that is written into the textbox at the left of this button
- *'Merge selected clusters'* - Merges all clusters that are currently selected in the prototype view area.
- *'Add empty cluster'* – Adds an empty cluster to the clustering
- *'Remove empty clusters'* – Removes empty clusters from the clustering
- 'Remove selected clusters' – Removes the clusters that are selected in the prototype view area.
- *'Re-cluster!'* – The clustering result is re-clustered by using their corresponding prototype images and an agglomerative hierarchical clustering scheme with the numbers of clusters to be generated specified in the editing box at the left of this button.
- *'Restore old clustering'* – The clustering result before the last editing operation can be restored

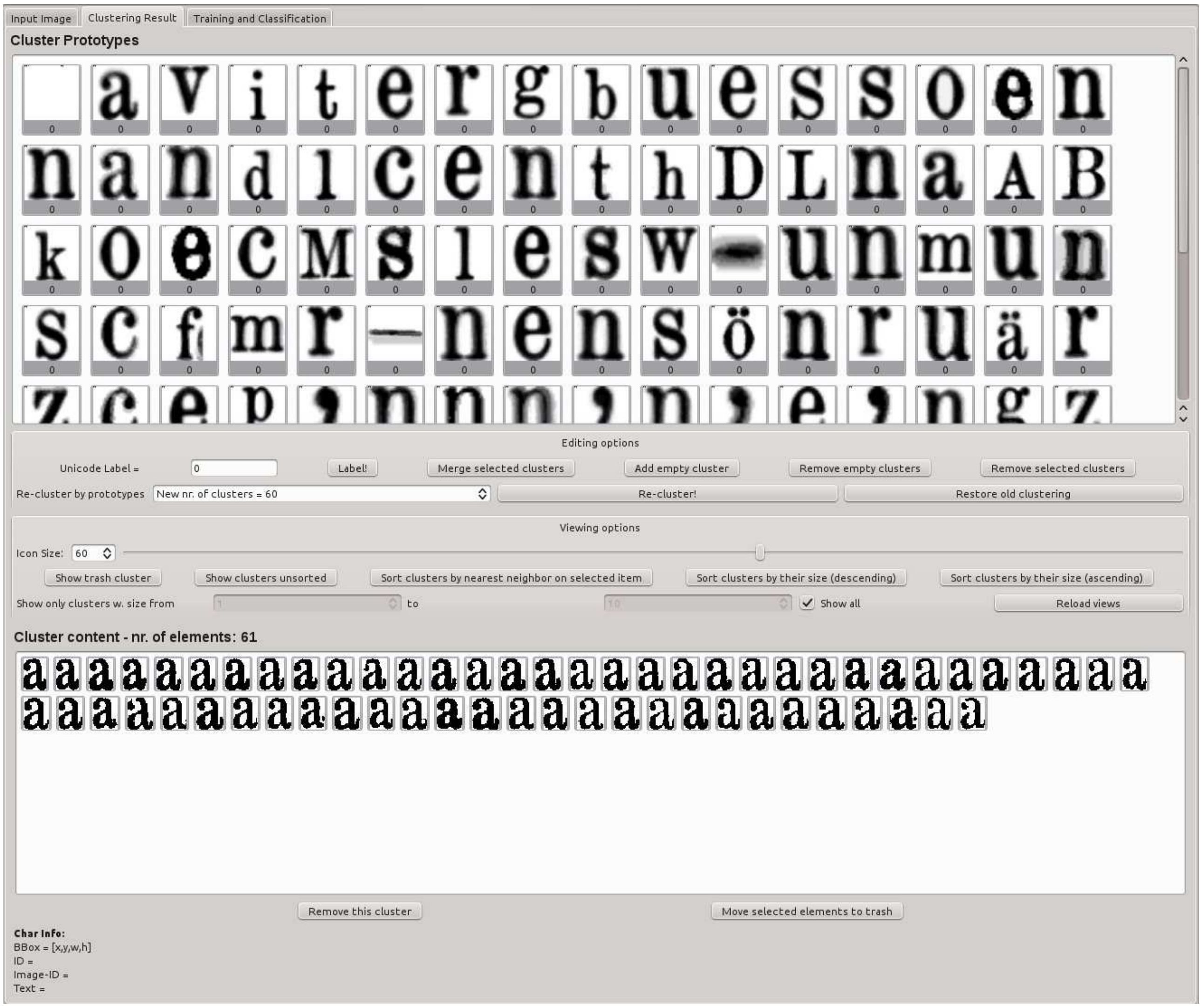
The buttons of the viewing options are:

- '*Icon Size*' – Use the slider or the spin-box to change the size of the cluster icons in the prototype view area.
- '*Show trash cluster*' – Shows the content of the trash cluster, which is a cluster where all characters tagged as noise are moved to.
- *'Show clusters unsorted'* – Shows all clusters unsorted, that is in order of their creation during the clustering process.
- *'Sort clusters by nearest neighbor on selected item'* - Performs a nearest neighbor search on

the prototype images for the selected cluster. This enables the user to quickly sort clusters which have a similar visual appearance and thus can be merged quickly.

- *'Sort clusters by their size (descending)'* and *'Sort clusters by their size (ascending)'* – Sorts clusters by their size in descending or ascending order respectively.
- *'Show only clusters with size from'* – Specify a size range for displaying clusters. Only clusters, which sizes fall into this range are displayed

Figures 9-10 show screenshots of the different parts of the clustering result view widgets.



Unicode Label = 0

Re-cluster by prototypes New nr. of clusters = 60

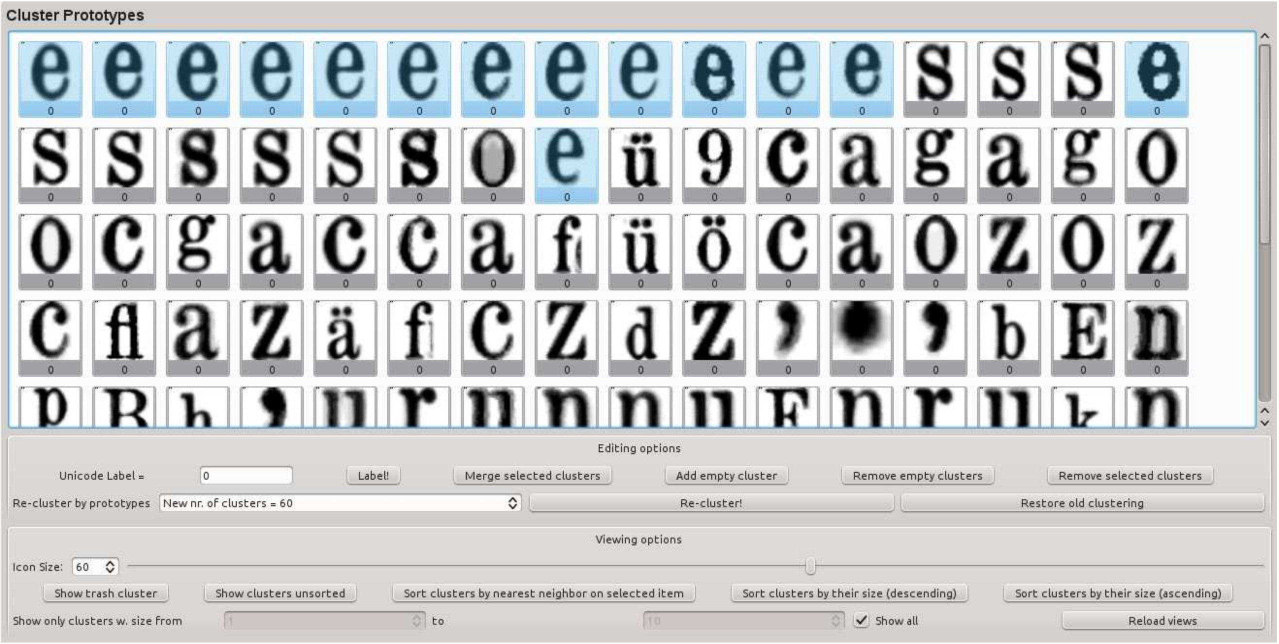
Icon Size: 60

Show only clusters w. size from 1 to 10 ☑ Show all

Cluster content - nr. of elements: 61

Char Info:
BBox = [x,y,w,h]
ID =
Image-ID =
Text =

Figure 8: Cluster visualization and editing widget. The widget is divided into two parts: cluster prototype visualization on the top and single cluster visualization at the bottom.



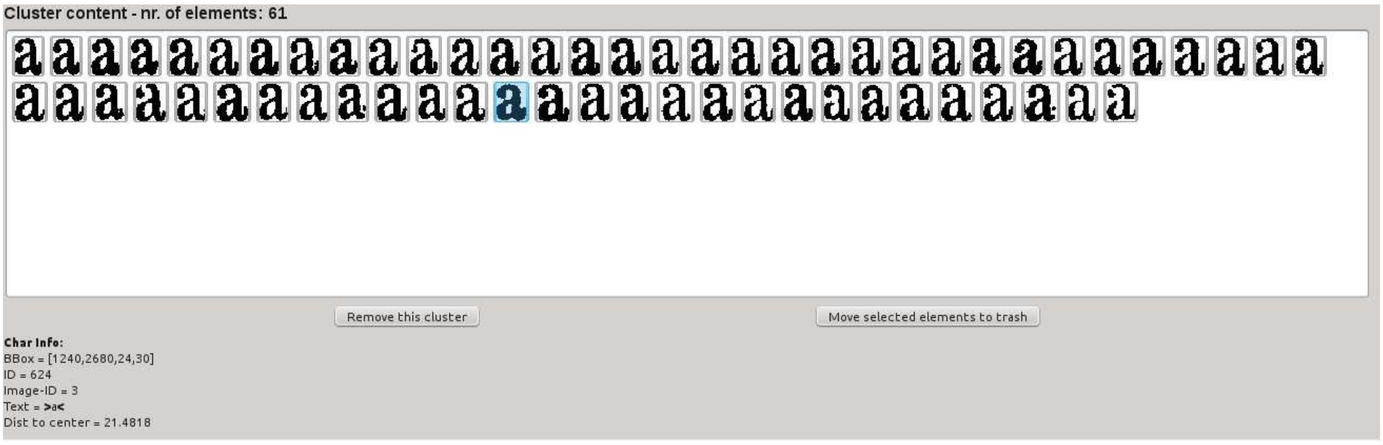
Unicode Label = 0

Re-cluster by prototypes: New nr. of clusters = 60

Icon Size: 60

Show only clusters w. size from 1 to 10 ☑ Show all

Figure 9: Cluster prototype widget with clusters sorted by nearest neighbor on an 'e' cluster. All similar clusters are now located in the beginning of the list and can therefore be selected and merged quickly.



Cluster content - nr. of elements: 61

Char Info:
BBox = [1240,2680,24,30]
ID = 624
Image-ID = 3
Text = >a<
Dist to center = 21.4818

Figure 10: Single cluster visualization widget. Characters that do not belong to this cluster can be selected and moved to another cluster using drag-and-drop operations.

9. Loading and saving clustering results

Once a clustering has been produced using a clustering algorithm and the post-editing functionality it can be saved using the '*Save Cluster Result...*' button located in the '*Clustering*' menu of the top menu bar. The user then has to specify a filename to produce an XML clustering result file. Clustering results can then also be loaded into the tool again using the '*Load Cluster Result...*' button located in the same menu.