Beyond OCR: Handwritten manuscript attribute understanding
He, Sheng

IMPORTANT NOTE: You are advised to consult the publisher’s version (publisher’s PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2017

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
He, S. (2017). Beyond OCR: Handwritten manuscript attribute understanding [Groningen]: University of Groningen

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
This chapter is an adaptation of the papers:
Sheng He, Petros Samara, Jan Burgers, Lambert Schomaker. – “Discovering visual element evolutions for historical document dating.”, Proc. of 15th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR2016), pp. 7-12, 22-26 October 2016, Shenzhen, China.

Chapter 6
Historical Manuscript Dating and Localization Using A Multiple-Label Clustering Algorithm

Abstract

Previous chapter proposes the contour and stroke fragments for historical manuscript dating, which based on binarized images. In this chapter, we propose a multiple-label guided clustering algorithm to discover the correlations between the concrete low-level visual elements in historical documents and abstract labels, such as date and location. Firstly, a novel descriptor, called Histogram of Orientations of Handwritten Strokes (HOHS or H\textsuperscript{2}OS), is proposed to extract and describe the visual elements, which is built on a scale-invariant polar-feature space. In addition, the Multi-Label Self-Organizing Map (MLSOM) is proposed to discover the correlations between the low-level visual elements and their labels in a single framework. Our proposed MLSOM can be used to predict the labels directly. Moreover, the MLSOM can also be considered as a pre-structured clustering method to build a codebook, which contains more discriminative information on date and geography. Experimental results on the Medieval Paleographic Scale (MPS) data set demonstrate that our method achieves state-of-the-art results.

6.1 Introduction

Many visual elements of images of the visual world can be correlated to a symbolic label (Lee et al., 2013). For example, visual elements from Google Street View Images contain geographical information (Doersch et al., 2012), visual elements of images of historical cars are correlated with temporal information (Lee et al., 2013) and visual elements in printed texts vary over different languages and scripts (Shijian and Tan, 2008; Ghosh et al., 2010). Finding the corresponding visual elements common to different labels can reveal the subtle difference between categories. Therefore, discovering the correlations between the visual elements style and their labels is very useful for resolving computer vision problems, such as localization (Doersch et al., 2012), dating (Lee et al., 2013), age estimation based on
face images (Guo et al., 2008), image classification (Singh et al., 2012) and script and font identification (Shijian and Tan, 2008; Ghosh et al., 2010).

In this chapter, we propose a novel descriptor, Histogram of Orientation of Handwritten Stroke (HOHS or H\textsubscript{2}OS), to extract and represent the visual elements (strokes or parts of the strokes) in historical documents. In contrast to existing features, such as the Histogram of Oriented Gradients (HOG) descriptor (Dalal and Triggs, 2005), our proposed H\textsubscript{2}OS is a scale-invariant descriptor which uses the stroke width as the scale factor. A weakly-supervised Self-Organizing Map (SOM) (Kohonen, 1988) method is proposed to introduce the label information in the self-organization process to discover the relationships of visual elements in each label space. Our proposed method is called Multi-Label Self-Organizing Map (MLSOM) which aligns the visual elements in multiple label spaces. We use the proposed H\textsubscript{2}OS feature and MLSOM method to answer these two questions for any query document: when was it written and where?

6.2 Histogram of Orientations of Handwritten Stroke Descriptor (H\textsubscript{2}OS)

6.2.1 Motivation

The Histogram of Oriented Gradients (HOG) descriptor (Dalal and Triggs, 2005) is widely used to describe the mid-level visual elements (Doersch et al., 2012; Lee et al., 2013; Singh et al., 2012; Juneja et al., 2013). However, the HOG does not contain any scale information and is always used in a multi-scale strategy. Applying the HOG in handwritten document images is computationally inefficient because the resolution of the document images is always high (300 dpi), leading to large image sizes. The SIFT (Lowe, 2004) is a scale invariant feature and is also widely used when addressing handwritten documents. However, SIFT features directly extracted from the entire document image are not discriminative because the keypoints are located not only on the strokes but also on the background or near the contour of strokes (Zhang and Tan, 2014), which introduces much noise. Usually, the SIFT detector is applied on the segmented word regions (Rusiñol and Lladós, 2014). However, word segmentation is a challenging problem in images of historical handwriting. In addition, the computation of the SIFT features in these images with a high resolution is also far from efficient.

In order to solve these problems, we propose a novel descriptor named Histogram of Orientations of Handwritten Stroke Descriptor (HOHS or H\textsubscript{2}OS for short), inspired by the Gradient Location-Orientation Histogram (GLOH) (Mikolajczyk and Schmid, 2005). There are three main steps to build the H\textsubscript{2}OS descriptor: (1) key-points selection; (2) scale-invariant log-polar space construction; (3) descriptor computation. Detailed information will be presented in the following sub-sections.
6.2. Histogram of Orientations of Handwritten Stroke Descriptor (H\textsubscript{2}OS)

Figure 6.1: The left figure shows the skeleton line detected in the handwritten character and the point \( p_1 \) is the fork point and the points \( p_2 \) and \( p_3 \) are the high curvature points which are the candidate points for the H\textsubscript{2}OS descriptor. The middle shows the stroke width distribution around the fork point. The scale factor is determined by the minimum value of this stroke width distribution which reflects the stroke width on the fork point. The right figure shows the log-polar space with 3 rings and 12 orientations. The circle in the center ring which is always filled by the ink pixels. The radius of the center ring is the scale factor of the proposed H\textsubscript{2}OS descriptor.

6.2.2 Key-points selection

We regard the structure points on the medial axis of handwritten strokes, such as the fork and high curvature points, as the key points. The structure points contain the topological information of the strokes and it has been shown in Chapter 4 that the regions around structure points contain discriminative information concerning writing styles. The procedure of the computation of structure points is as follows. First, the handwritten document is binarized and the medial axis (also known as skeleton line) is extracted by thinning methods. Then the fork points are detected by (Liu et al., 1999) and the high curvature points are detected by the method proposed in Chapter 4. The left figure of Fig. 6.1 shows an example of fork point and high curvature points detected on the skeleton lines.

6.2.3 Scale-invariant log-polar space construction

The text in handwritten document images often has very inconsistent character sizes and document images are often digitized with different resolutions, which requires the use of a scale-invariant descriptor. In this chapter, we consider the stroke width as the determinant for the scale factor because, usually one and the same or at least a similar writing instrument was used, (e.g., a quill), yielding a typical average stroke width. Given the key point \( p_i \), the stroke width can be estimated using the method proposed in Chapter 4 as follows. First, a stroke length distribution (see the middle figure of Fig. 6.1) is built by computing the stroke length \( \text{len}(\theta) \) from the key point \( p_i \) to the stroke boundary at each direction \( \theta \) from 0 to \( 2\pi \) using the method proposed in (Brink et al., 2012). Secondly, the stroke width \( w_{\text{stroke}}(p_i) \) at
the point \( p_i \) is estimated as the minimum value in the stroke length distribution:

\[
w_{\text{stroke}}(p_i) \approx 2 \times \min_\theta \text{len}(\theta)
\]  

(6.1)

Given the key point \( p_i \) and the scale factor \( w_{\text{stroke}}(p_i) \), a log-polar space can be built on \( p_i \), which is the popular structure of the existed local descriptors, such as Shape Context (Belongie et al., 2002), Self-Similar Descriptor (SSD) (Shechtman and Irani, 2007) and Histogram of Orientation Shape Context (HOOSC) (Roman-Rangel et al., 2011). The size of the log-polar space is determined by the number of angular intervals \( N_{\text{ang}} \) and the number of distance intervals \( N_r \). The distance intervals is equal to the half of the stroke width \( w_{\text{stroke}}(p_i)/2 \) in the log-polar space. An example of the log-polar space is shown in the right figure of Fig. 6.1. From the right figure of Fig. 6.1 we can see that the center ring in the log-polar space is always filled by the stroke ink and contain very little information. Therefore, we discard this region and, finally, there are \( N_{\text{ang}} \times N_r \) bins in our H₂OS descriptor.

### 6.2.4 Descriptor computation

For a given input handwritten image \( I \), we first compute the orientation map \( G_\theta \) on the orientation \( \theta \) following (Tola et al., 2010) as:

\[
G_\theta = \left( \cos \theta \frac{\partial I}{\partial x} + \sin \theta \frac{\partial I}{\partial y} \right)^+
\]  

(6.2)

where \((\cdot)^+\) is the operator such that \((a)^+ = \max(a, 0)\) to keep the only positive values to preserve the polarity of intensity changes (Tola et al., 2010; Huang et al., 2014). The gradient orientations in each region of the log-polar space are quantized in \( N_\theta \) bins. In order to eliminate the quantized errors and avoid abrupt changes in the orientation map, a Gaussian kernel \( G_\sigma \) with the standard deviation \( \sigma \) is introduced to convolve the orientation map to obtain a smooth version:

\[
\tilde{G}_\theta = G_\sigma * G_\theta
\]  

(6.3)

Finally, the histogram of orientation in each region of the log-polar space on the point \( p_i \) is computed as:

\[
h'_r = [\tilde{G}'_1, \ldots, \tilde{G}'_{N_\theta}]
\]  

(6.4)

where \( r \) denotes the index of region in the log-polar space, and \( \tilde{G}'_i \) is the integrated value of the orientation map, \( 1 \leq i \leq N_\theta \), in the region \( r \). The H₂OS descriptor is obtained by concatenating all \( N_{\text{ang}} \times N_r \) histograms of orientation in the log-polar space, yielding a local feature vector with \( N_{\text{ang}} \times N_r \times N_\theta \) dimensions. The descriptor is normalized dependent on each distance interval (or each ring) inspired by HOOSC.
6.3. Multi-Label Self-Organizing Map (MLSOM)

6.2.5 Descriptor analysis

The computation of the proposed H$_2$OS descriptor is very similar to SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005), and can be regarded as an extrapolation of GLOH ( Mikolajczyk and Schmid, 2005) to handwritten document images. Therefore, it inherits all their properties. However, our proposed H$_2$OS descriptor is built on a scale-invariant log-polar space using the stroke width as the scale factor, thereby making the H$_2$OS scale invariant.

In addition, the key points of our proposed H$_2$OS are always located inside the ink strokes, capturing the stroke structure information instead of the textural information of handwritten text.

The differences between our proposed H$_2$OS descriptor and the junction descriptor proposed in Chapter 4 are that: (1) The junction descriptor is a binarized-based descriptor while the proposed H$_2$OS is a gradient-based feature which contains more rich local information than the junction descriptor. (2) The proposed H$_2$OS descriptor has a larger support region than the junction feature, as show in Fig. 6.1 which could describe more context information around the key points. (3) The H$_2$OS descriptor captures the contrast and orientation information on the ink contours of handwritten images, which is more robust when dealing with the poor quality images.

The number of rings $N_r$ of the H$_2$OS should be selected carefully, because if $N_r$ is too small, the H$_2$OS will focus on local regions which contain little information, while if $N_r$ is too large, the H$_2$OS will contain a lot of background noise. This can be observed in Fig. 6.2. We find that H$_2$OS almost always covers a meaningful visual element when $N_r = 3$, and this value is considered to build our H$_2$OS descriptor. We set the number of angular intervals of the log-polar space $N_{ang} = 12$ and the number of bins of the quantized orientations $N_{\theta} = 8$. Finally, the dimension of the H$_2$OS is $12 \times 3 \times 8 = 288$.

6.3 Multi-Label Self-Organizing Map (MLSOM)

6.3.1 Motivation

Given the visual elements extracted by the proposed H$_2$OS feature, one possible way to discover the correlations between the visual elements and the labels, such as the year and the city, has been proposed in (Lee et al., 2013; Doersch et al., 2012). The general procedure...
Historical Manuscript Dating and Localization Using MLSOM

is: (1) use the unsupervised cluster method (such as the $k$-means) to obtain the clusters in each label space; (2) select the discriminative clusters which correlate with their labels using an exhaustive-search method on the whole data set; and (3) train an SVM classifier for each selected cluster and find the correspondences across the entire data set. The main disadvantage of this approach is that it cannot be directly used in multiple-label spaces at the same time.

In this chapter, we integrate these three steps into one framework, inspired by the property of the Self-Organizing Map (SOM) neural network which can preserve the topological properties of the input space using a neighborhood function. Moreover, we can integrate more than one label in the proposed framework and visual elements can align in multiple label spaces simultaneously.

6.3.2 MLSOM configuration

In the traditional unsupervised SOM, the dimension of the grid is usually low (1D or 2D). However, in the MLSOM we assume that each dimension corresponds to one label space. If there are $L$ labels for each visual element, the dimension of the MLSOM will be $L + 1$, in which the extra dimension is the visual element space itself to preserve the topology of visual elements. Fig. 6.3 gives an example of MLSOM with 2 label spaces. Each node in the MLSOM neural network is connected to neighbor nodes in each label space (see Fig. 6.4).

6.3.3 MLSOM Training

Given the labeled training visual elements $v_i = (x_i, Y^L)$ where $x_i \in \mathbb{R}^d$ is the low-level representation of the $i$-th visual element, $Y^L = \{y_i^1, y_i^2, \ldots, y_i^L\}$ is the label space and $L$ is the number of labels, our aim is to align these visual elements in the $L + 1$ spaces (with the extra
6.3. Multi-Label Self-Organizing Map (ML SOM)

Visual element space). In the traditional SOM model, there are two main training stages: the competitive stage and the cooperative stage. We adapt these two stages as follows to train our proposed MLSOM neural network.

**Competitive stage:** The basic idea of the competitive learning is that “only one cell or local group of cells at a time gives the active response to the current input [Kohonen 1988].” In the competitive stage, the winner neuron is the one whose weight is most similar to the input vector, which is also known as the Best Matching Unit (BMU).

\[
q^* = \arg \min_q \{ D(x_i^L, w_q^L) \} \tag{6.5}
\]

where \( w_q^L \) is the \( q \)-th neuron in the MLSOM with the same label as the training sample \( x_i^L \), \( q^* \) is the index of the winner neuron in the extra visual element space, \( 1 \leq q^* \leq v \) where \( v \) is the dimension of visual element space and \( D \) is a distance function. The MLSOM has \( L + 1 \) spaces with an extra visual element space, thus the searching is performed only on this extra visual element space, making sure that the BMU neuron has the same labels with the training sample. (Note that there is no label for the extra visual element space.)

**Cooperative stage:** Any neurons who are the neighbors of the BMU are updated their weights to preserve the topological order, by defining a neighborhood set \( N_q^* \). The learning process is defined as:

\[
w_q(t + 1) = \begin{cases} 
    w_q(t) + \eta(t)(w_q(t) - x_i) & \text{if } q \in N_q^*(t) \\
    w_q(t) & \text{if } q \notin N_q^*(t)
\end{cases} \tag{6.6}
\]
Assume that the black neuron is the BMU, then the neighbors of the BMU are the neighbors in the coordinates of the MLSOM if the MLSOM network is initialized with the label order. For example, the neighbors of the BMU are the connected left and right neurons (the gray ones).

where $t = [0, T]$ is the epoch counter, $T$ is the number of training iterations and the $\eta(t)$ is the learning rate which is defined, following (Schomaker and Bulacu, 2004), as:

$$\eta(t) = \left( \frac{\eta_T^{1/s} - \eta_0^{1/s}}{T} + \frac{\eta_0^{1/s}}{s} \right)^s$$  \hspace{1cm} (6.7)

where $s(>0)$ is the steepness factor, $\eta_0$ and $\eta_T$ are the starting and ending values. This is a decreasing function and the maximum value $\eta_0$ decreases to $\eta_T$.

The most important part of MLSOM is the definition of the neighborhood set $N_{q^*}$. Traditionally, a Gaussian function is widely adopted as the neighbor function in the unsupervised SOM, which finds the coordinate neighbors around the BMU in the SOM neural network. However, in the proposed MLSOM, we want to discover the connections in the label spaces and the coordinate neighbors are not always the neighbors in the label space.

There are two types of visual element label space: the ordered label space and non-ordered label space. In the ordered label space, the labels have an inherent order. For example, $y_i+1$ always follows $y_i$. There are many different ordered label spaces, such as a time sequence, the year (key year) labels and people’s ages. Fig. 6.5 shows an example of a key year label space, which is an ordered label space. If we initialize the MLSOM with the same order as labels, the neighbors of the BMU in the label space is the same as the neighbors of the BMU in the coordinates of the MLSOM neural network, which can be computed as:

$$N_{q^*}^y(t) = [q^*y - r(t), q^*y + r(t)]$$  \hspace{1cm} (6.8)

here $y_i$ is the index of label space and $r(t)$ is the spatial resolution of the neighbors at epoch $t$, which can be determined by Eq. 6.7. Note that $r(t)$ is the decreasing function, which indicates that at the beginning of the training, the SOM aims to build a general connection in a large scope in the $y_i$ label space and at the end of the training $r(t)$ tends to zero and the MLSOM is finely tuned to preserve the discriminative information.

In the real-world, most labels are non-ordered, such as the categories of animals or cities (see an example in Fig. 6.6). In this case, the neighbors of the BMU in the coordinate of the MLSOM neural network can not reflect the neighbors in the label space and the Eq. 6.8 can not be used as the neighbor function. In this chapter, we assume that the BMU neuron is fully connected to other neurons in the non-ordered label space and the neighbors of the BMU are determined as the top $N_{top}(t)$ similar neurons according to the distance function $D$. The
6.3. Multi-Label Self-Organizing Map (MLSOM)

Figure 6.6: An example of animal label space (non-ordered label space). Assume that the neurons represent the visual elements about head of animals and the black neuron is the BMU, then the neighbors of BMU should be determined by the similarity between the BMU and other neurons. For example, the top 2 neighbors of the head of the cat should be the heads of the dog and the tiger (according to the similarity of visual elements of the animal’s heads). In fact, all the neurons have no semantic meanings and thus similarity is computed on the low-level feature space.

$N_{top}(t)$ is the spatial resolution in this non-ordered label space and can also be determined using Eq. 6.7. Fig. 6.6 shows an example of the neighbors of the BMU in the non-ordered space.

### 6.3.4 Learning from neighbors

Zero-shot learning aims to learn the labels of an image, in the case where no visual examples of that labels are available during training (Mensink et al., 2014). The relationships between the annotated classes and the unseen classes are built usually on a high level, such as the attributes (Lampert et al., 2014) or co-occurrences of visual concepts (Mensink et al., 2014). This information is often learned from a large labeled data set. The assumption of this chapter is that the visual elements of a certain class have a subtle but consistent difference with its neighbors, which can be captured by the proposed MLSOM neural network. Therefore, our method can also be used for zero-shot learning by learning from the neighbors. In the earlier training stage, the neurons in the MLSOM are updated not only by the training visual elements when they are the winner neurons, but also by the visual elements when their neighbor neurons are the winner neurons. Therefore, each neuron in the MLSOM also learns from their neighbors until the window size closed to zero. If there is no training sample in a certain label space, the corresponding neurons will be updated by the neighbors and thus contain the average information of their neighbor neurons.
6.3.5 MLSOM Voting

Each neuron in the trained MLSOM neural network carries labels. Therefore, the trained MLSOM can be directly used to predict the labels of the test images. The visual elements $v_i$ from the test image $I$ can be mapped into the learned MLSOM space to form a histogram $h^{L+1}$, which is similar with the bag-of-word model [Fei-Fei and Perona 2005]. The probability $p(y'_l)$ that the test image $I$ belongs to the label $y'_l$ in $l$-th label space is estimated by:

$$p(y'_l) = \frac{\sum_{q=0, q \neq l}^{L} h_q(i)}{\sum_j \sum_{q=0, q \neq l}^{L} h_q(i)}$$  \hspace{1cm} (6.9)

here $\sum_{q=0, q \neq l}^{L} h_q(i)$ counts the number of occurrences of corresponding visual word in MLSOM along other label spaces except $l$-th label space and the $\sum_j \sum_{q=0, q \neq l}^{L} h_q(i)$ is the total number of occurrences. Finally, the test image is assigned to the label with the highest probability:

$$y'(I) = \arg \max_{y'_l} \{ p(y'_l) \}$$  \hspace{1cm} (6.10)

6.4 Experiments

6.4.1 Document stroke-shape elements

Generally, the stroke structures are very often repeated in handwritten texts of historical documents (Fig. 6.7), because the number of letters in an alphabet is limited and their appearances in the feature space are quite similar due to style considerations in the writer. Therefore, regarding them as separate visual elements makes inefficient use of the available information. Our goal in this section is to detect a set of primary visual elements that represents the wide variety of stroke structures in each historical document, which contains the discriminative information concerning writing style. We call the detected primary visual elements Stroke-Shape Elements. We assume that each Stroke-Shape Element represents a style element. The stroke-shape elements are clusters of these sampled patches generated by the $k$-means, algorithm from the $H_2OS$ descriptor using the $\chi^2$ distance. Typical individual cluster centroids are illustrated in Fig. 6.8. The main reason we use the Stroke-Shape Element instead of the sampled patch instances in the document is to avoid the effects of an unbalanced number of instances for each Stroke-Shape Element in the voting stage. For example, a large number of instances of the character ‘e’ with the same writing style will lead to a large number in the corresponding bin in the voting histogram. However, if we use the Stroke-Shape Element representation, there is only one writing style element contributing to the voting histogram.
6.4.1 Experimental setup

There are two label spaces: The key-year space and the city space and one extra visual-element space for historical document dating and localization. The key-year space is an ordered label space and the size is 11, while the city space is a non-ordered label space and the size is 4. The size of the extra visual element \( v \) should be set manually. The parameters in the training of the MLSOM neural network are set as follows: the starting value of the learning rate \( \eta(t) \) is set \( \eta_0 = 0.5 \) and the end is set \( \eta_T = 0.005 \); for the spatial resolution of the neighbors \( r(t) \), the starting value is set \( r(0) = 5 \) and the end is set \( r(T) = 0 \). A steepness factor of \( s = 5 \) was used to compute the learning rate and neighbor size, following [Schomaker].
Table 6.1: The average and standard deviation MAEs for dating with different values of the number of clusters $k$ and the size of visual elements spaces $v$.

<table>
<thead>
<tr>
<th>System</th>
<th>$v$</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>$k$</td>
<td>50.0</td>
<td>35.2</td>
<td>30.9</td>
<td>28.2</td>
<td>28.0</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>35.4</td>
<td>30.4</td>
<td>27.3</td>
<td>25.9</td>
<td>25.9</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td>37.3</td>
<td>33.1</td>
<td>28.5</td>
<td>27.1</td>
<td>26.2</td>
</tr>
<tr>
<td>300</td>
<td></td>
<td>38.1</td>
<td>33.6</td>
<td>29.0</td>
<td>27.5</td>
<td>26.8</td>
</tr>
<tr>
<td>400</td>
<td></td>
<td>39.8</td>
<td>33.4</td>
<td>29.7</td>
<td>28.1</td>
<td>26.9</td>
</tr>
</tbody>
</table>

and Bulacu (2004) and the number of training iterations is $T = 500$.

6.4.3 MLSOM voting results

There are two parameters that need to be optimized: the number of clusters $k$ for the stroke shape elements and the size of visual elements spaces $v$. We used a grid search method to find the appropriate values and Table 6.1 shows the performances of historical document dating with different values. From Table 6.1, we can see that increasing the number of stroke shape elements $k$ does not improve the performance, which results from the fact that the number of different writing styles is limited in one document and a large value of $k$ introduces more noise. The best value of $k$ is 100 in our experiments for different values of $v$. Conversely, the performance is better when the value $v$ is higher. However, a high value of $v$ also needs a large memory and long computing time. In our experiments, we set the value of $v$ to 300, which achieves the best performance (the MAE is 25.9 years). The localization precision is shown in Table 6.2. The performance is higher when $k$ is smaller and $v$ is higher, which is the same as the dating performance in Table 6.1. The best performance for localization is achieved when $k = 50$ and $v = 400$ and 83.8% documents are correctly localized.

Table 6.3 shows the MAEs of the proposed method compared with our previous methods, as well as a random guess. The method which is from the Monk system (Van der Zant et al., 2008) used the human labeled characters for dating and the work in (He et al., 2014) used global and local regression methods with the Hinge and Fraglets (Bulacu and Schomaker, 2007) features. Our method improves the performance and the best MAE is 25.9 years which is almost 10 years lower than previous results. Fig. 6.9 shows the CS measures for different methods. We can observe that the proposed method also improves the score for lower error level, e.g., $\alpha \leq 25$ years. For example, 46.2%$(\pm 7.2)$ documents are estimated correctly by the proposed method, which is higher than the 22.5% given by the method in (He et al., 2014). When the year error level is 25 years, the proposed method could improve the

http://application02.target.rug.nl/monk/Overslag/date-histogram-MPS.html
Table 6.2: The average and standard deviation precisions (%) for localization with different values of the number of clusters $k$ and the size of visual elements spaces $v$.

<table>
<thead>
<tr>
<th>System</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>77.0%±3.6</td>
<td>80.2±4.8</td>
<td>82.4±4.0</td>
<td>83.2±3.4</td>
<td><strong>83.8±4.1</strong></td>
</tr>
<tr>
<td>100</td>
<td>75.8%±4.5</td>
<td>78.7±4.5</td>
<td>80.6±4.8</td>
<td>81.8±4.7</td>
<td>82.8±5.1</td>
</tr>
<tr>
<td>200</td>
<td>76.8%±4.2</td>
<td>78.0±5.2</td>
<td>81.3±5.8</td>
<td>81.3±5.8</td>
<td>81.8±4.4</td>
</tr>
<tr>
<td>300</td>
<td>75.1%±3.1</td>
<td>74.9±6.2</td>
<td>78.9±5.6</td>
<td>81.1±5.3</td>
<td>80.8±5.3</td>
</tr>
<tr>
<td>400</td>
<td>75.0%±4.4</td>
<td>75.9±4.5</td>
<td>78.1±6.2</td>
<td>79.4±5.3</td>
<td>81.4±4.7</td>
</tr>
</tbody>
</table>

Table 6.3: The MAEs, Standard Deviations (STD) and CSs($\alpha=25$) on our database with different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>CS($\alpha=25$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>85.3±58.5</td>
<td>25.7%</td>
</tr>
<tr>
<td>Monk (Van der Zant et al., 2008)</td>
<td>36.0±20.6</td>
<td>-</td>
</tr>
<tr>
<td>Study (He et al., 2014)</td>
<td>35.4</td>
<td>63.5%</td>
</tr>
<tr>
<td>MLSOM voting</td>
<td>25.9±4.5</td>
<td>73.7%</td>
</tr>
</tbody>
</table>

accuracy by 10.2% (see the last column in Table 6.3).

Fig. 6.10 provides the results of the localization precisions with the year error levels from 0 to 100 years. 41.5%(±6.3%) documents are estimated correctly with the date and local information. When the year error level is 25 years, the precision of the localization is 63.2%(±5.1%). The results demonstrate that dating and localizing simultaneously is more difficult than dating and localizing separately. Table 6.4 shows the confusion matrix for the historical document localization. From the table we can observe that the documents from Leuven and Groningen are quite easy to localize than those from Arnhem and Leiden. The precision for Leiden is only 68.4%(±9.2%) and the documents from Leiden are easy to be estimated as documents from Arnhem and Leuven.

6.4.4 Dating by classification

The dating problem can be considered as a classification problem because the document distribution in the considered period has a obvious borders between the nearby key years in the MPS data set according to the domain experts (paleographers). We assume that all the documents from the same key year form a class and there are 11 classes (11 key years) in our MPS data set. We train 11 classifiers using a linear SVM with the one-versus-all strategy. The parameter $C$ is estimated by a grid search method in the range of $\{2^{-16}, 2^{-15}, \ldots, 2^{15}, 2^{16}\}$.
The query document is assigned to the key year with the maximal SVM score among the 11 trained classifiers.

The proposed MLSOM is also a cluster method which can be used to train the codebook. We map the extracted patches described by $H_2OS$ features to the codebook to compute the representation of the documents using the bag-of-word model, which is denoted as $H_2OS_{mlsom}$. We believe that the codebook trained by the MLSOM is more discriminative than the traditional SOM methods because it contains the label information. In order to evaluate this observation, we train a codebook using the traditional SOM method with the same data and the same parameters, which is denoted as $H_2OS_{som}$.

We also compare our method with the existing features used for writer identification in handwritten documents, which can be typically divided into two categories: textural-based and grapheme-based features. Two textural-based, such as Hinge and Quill, and two
6.4. Experiments

Table 6.4: The accuracy (%) of the localization confusion matrix among the four cities.

<table>
<thead>
<tr>
<th>City</th>
<th>Arnhem</th>
<th>Leiden</th>
<th>Leuven</th>
<th>Groningen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arnhem</td>
<td>86.2±3.0</td>
<td>7.6±3.3</td>
<td>2.7±1.2</td>
<td>3.5±1.2</td>
</tr>
<tr>
<td>Leiden</td>
<td>18.1±5.6</td>
<td>68.4±9.2</td>
<td>10.2±6.7</td>
<td>3.3±0.5</td>
</tr>
<tr>
<td>Leuven</td>
<td>0.4±0.6</td>
<td>2.2±2.8</td>
<td>97.0±3.5</td>
<td>0.4±0.5</td>
</tr>
<tr>
<td>Groningen</td>
<td>4.6±3.9</td>
<td>2.1±1.8</td>
<td>1.3±0.6</td>
<td>92.0±4.5</td>
</tr>
</tbody>
</table>

grapheme-based features, such as Junclets and Strokelets, are selected in the experiments, which are:

**Hinge**: The Hinge is a texture level feature ([Bulacu and Schomaker, 2007] [Schomaker and Bulacu, 2004]), which captures the slant and curvature information of the handwritten ink trace. The Hinge feature is a joint probability distribution of the orientations of the two edge fragments constituting the legs of an imaginary hinge. The two parameters of the Hinge feature, the number of angle bins \( p \) and the leg length \( q \), are set to \( p = 40 \) and \( q = 20 \).

**Quill**: The quill feature was designed to capture the property of the capillary-action of writing instruments, such as the “quill pen” which were used until the 19th century ([Brink et al., 2012]). The Quill feature is a probability distribution of the relation between the ink direction and the ink width. The parameters are set following the original paper ([Brink et al., 2012]).

**Junclets**: The junction regions are important visual elements in handwritten documents which reflecting the writing style. The junction feature proposed in Chapter 4 computes the distribution of the stroke length on the junction point in a polar space. The Junclets is the probability-density function of the junctions based on a trained codebook with the size of 625.

**Strokelets**: The connected components of the handwritten text are segmented into sub-strokes on the fork points and the Polar Stroke Descriptor which is similar with the junction feature is used to describe the sub-strokes ([He and Schomaker, 2015]). The Strokelets is also a probability-density function of the sub-strokes based on a trained codebook with the same size of Junclets.

After computing the feature representations of documents in the MPS data set, we perform dating in two ways: Dating while excluding writer duplicates versus inclusion of writer duplicates. The exclusion of writer duplicates enforces a style-based dating, as opposed to a dating result which can be attributed to writer identification. In the following sections, we compare the performance of the proposed \( H_2 OS_{tom} \) and \( H_2 OS_{mltom} \) representations, as well as the Hinge, Quill, Junclets and Strokelets features in the same experimental setting.
Table 6.5: The dating performance with different configurations when excluding writer duplicates.

<table>
<thead>
<tr>
<th>Feature</th>
<th>MAEs</th>
<th>CS((\alpha=25))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge</td>
<td>22.1±2.9</td>
<td>80.6%±3.1</td>
</tr>
<tr>
<td>Quill</td>
<td>23.7±2.9</td>
<td>80.5%±3.1</td>
</tr>
<tr>
<td>Junclets</td>
<td>21.7±3.7</td>
<td>79.4%±4.4</td>
</tr>
<tr>
<td>Strokelets</td>
<td>19.4±2.7</td>
<td>83.1%±2.6</td>
</tr>
<tr>
<td>H(^2)OS(_{som})</td>
<td>25.2±3.2</td>
<td>76.4%±3.7</td>
</tr>
<tr>
<td>H(^2)OS(_{mlsom})</td>
<td>15.9±2.9</td>
<td>85.4%±3.7</td>
</tr>
</tbody>
</table>

**Dating results when excluding writer duplicates**

In the MPS data set, the writers of several documents are known and the writers of the rest of the documents are unknown. In order to avoid effectively practicing writer identification when dating, we carefully and randomly split the data set into training (70%) and testing (30%) sets to make sure that the same writer never appears in both the training and the testing set, which means that all documents produced by the same hand should be only in the training set or only in the testing set. The experiment is repeated 20 times and the average results with the standard deviation are reported.

Table 6.6 shows the MAEs of different methods. We can see that our proposed H\(^2\)OS\(_{mlsom}\) achieves the best performance in terms of both MAE and CS(\(\alpha=25\)). The performance of the H\(^2\)OS\(_{mlsom}\) method using the codebook trained by the proposed MLSOM method is much better than the H\(^2\)OS\(_{som}\) which uses the traditional cluster method to train the codebook. The H\(^2\)OS\(_{mlsom}\) reduces the MAE by 9.3 years, which demonstrates that the proposed MLSOM codebook contains more discriminative information than the codebook trained by traditional cluster methods.

The performance of the proposed H\(^2\)OS\(_{mlsom}\) outperforms other methods. In the existing features, the Strokelets achieves the best performance. However, the MAE of the Strokelets is 19.4 years, which is lower than the proposed H\(^2\)OS\(_{mlsom}\) by 3.5 years.

**Dating results when including writer duplicates**

In this section, we conduct the experiment of the dating on the MPS data set by randomly splitting the documents into training (70%) and testing (30%) sets without considering whether the documents from the same hand appear in the training set or in the testing set. Table 6.6 shows the MAEs and the CS with 25 error level for different methods. When using the proposed H\(^2\)OS feature, the MAE of the H\(^2\)OS\(_{mlsom}\) is higher than the MAE of the H\(^2\)OS\(_{som}\) by 6.7 years. It also improves the measure CS(\(\alpha=25\)) from 85.4% to 91.4%, which means 91.4% documents are correctly estimated with the error equal or less than 25 years by the H\(^2\)OS\(_{mlsom}\). Compared to other features, our proposed H\(^2\)OS\(_{mlsom}\) achieves the best performance in terms of MAEs (9.1 years) and CS(\(\alpha=25\)) (91.4%). The results are very good, however, entailing a contamination over writer identity which may, or may not bother
6.4. Experiments

Table 6.6: The performance of the dating with different features when including writer duplicates.

<table>
<thead>
<tr>
<th>Feature</th>
<th>MAEs</th>
<th>CS(α=25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge</td>
<td>12.2±0.9</td>
<td>89.6%±1.3</td>
</tr>
<tr>
<td>Quill</td>
<td>12.1±1.0</td>
<td>89.5%±1.3</td>
</tr>
<tr>
<td>Junclets</td>
<td>12.4±0.7</td>
<td>88.4%±1.4</td>
</tr>
<tr>
<td>Strokelets</td>
<td>11.4±0.9</td>
<td>89.4%±1.7</td>
</tr>
<tr>
<td>H₂OS_{som}</td>
<td>15.8±1.8</td>
<td>85.4%±1.6</td>
</tr>
<tr>
<td>H₂OS_{mlsom}</td>
<td>9.1±0.8</td>
<td>91.4%±1.5</td>
</tr>
</tbody>
</table>

Table 6.7: The average MAEs and the number of documents in three periods.

<table>
<thead>
<tr>
<th>Feature</th>
<th>1300-1375</th>
<th>1400-1475</th>
<th>1500-1550</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge</td>
<td>17.7±6.9</td>
<td>11.8±2.4</td>
<td>8.8±2.1</td>
</tr>
<tr>
<td>Quill</td>
<td>16.2±6.6</td>
<td>11.6±2.4</td>
<td>9.5±2.2</td>
</tr>
<tr>
<td>Junclets</td>
<td>15.0±4.9</td>
<td>13.1±2.3</td>
<td>9.3±2.0</td>
</tr>
<tr>
<td>Strokelets</td>
<td>16.5±5.4</td>
<td>10.9±2.3</td>
<td>7.9±1.8</td>
</tr>
<tr>
<td>H₂OS_{som}</td>
<td>22.7±6.2</td>
<td>11.4±1.9</td>
<td>10.9±1.9</td>
</tr>
<tr>
<td>H₂OS_{mlsom}</td>
<td>11.6±4.2</td>
<td>7.2±1.4</td>
<td>6.7±1.5</td>
</tr>
<tr>
<td>Number of documents</td>
<td>505</td>
<td>1490</td>
<td>873</td>
</tr>
</tbody>
</table>

Table 6.7 shows the MAEs of three different periods: 1300-1375, 1400-1475 and 1500-1550. From the table we can see that dating documents from the period of 1500-1550 is much easier than documents from the other two periods.

Stability of codebooks trained with writer-related bias

In this section, we evaluate the performance of dating with the different MLSOM codebooks trained by documents from a subset of writers in each key year in order to evaluate the performance with writer-related bias. Table 6.9 shows the performance with codebook trained by documents from all writers (denoted as Codebook _wr.incl._) and with codebook trained by documents only from a subset of writers (only one-fourth writers are involved in this experiment, denoted as Codebook _wr.excl._). From the table we can see that including all writers in the codebook training provides slightly better results. This shows that handwritten
patterns written by different writers in the MPS data set are variable and it is better to train the codebook with all writers.

Dating results with different image quality

Some documents in the MPS data set have heavy degradation (low quality, see Fig. 6.11). In order to evaluate the performance with different image quality, we split all documents in MPS into high-quality and low-quality sets. We manually select 109 documents with very high quality and 118 documents with very low quality to train a kernel SVM to predict the quality of all documents on the MPS data set. Seven features are used to represent the documents as follows: f1: the absolute distance between the mean values of the Gaussian distributions of the ink and background pixels and f2: the ratio between the standard deviations of these two Gaussian distributions; f3: the distribution of the number of connected components; f4: the entropy of the distribution of the contour length of connected components; f5: the entropy of the distribution of the stroke width estimated by the method (Brink et al., 2012); f6: the ratio of the uniform LBP pattern and non-uniform ones (Ojala et al., 2002); f7: the entropy of the distribution of the line length computed by the fast line detector (LSD) (Von Gioi et al., 2010). In practice, we have found that these seven features work very well and the accuracy is over 90% on the MPS data set.

Fig. 6.9 shows the performance with different image qualities. From the figure we can see that (1) the performance of all features shows the same trend as the one without considering the image quality: Quill < $H_2OS_{com}$ < Hinge < Junclets < Strokelets < $H_2OS_{mlsom}$ according to their performance and (2) the results on the high-quality set are better than the results on the low-quality set for all features.

We also conduct the experiment of dating by using a high-quality set for training and a low-quality set for testing, and vice versa. Table 6.10 shows the MAEs of different features and we can see that the results of using low-quality set for training are better than using high-quality set. One important observation that can be made is that our proposed $H_2OS_{com}$ and $H_2OS_{mlsom}$ methods are much more stable when dealing with image qualities, achieving the minimum differences between the MAEs when using different image qualities for training.
6.4. Experiments

Table 6.9: The MAEs of different features with different image quality. high.incl. and high.excl. mean the performance with high quality when including writer duplicates (incl.) and excluding writer duplicates (excl.), respectively. low.incl. and low.excl. mean the performance with low quality.

<table>
<thead>
<tr>
<th>Features</th>
<th>high.incl.</th>
<th>high.excl.</th>
<th>low.incl.</th>
<th>low.excl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge</td>
<td>10.9±0.7</td>
<td>19.7±2.4</td>
<td>13.6±1.3</td>
<td>25.9±2.4</td>
</tr>
<tr>
<td>Quill</td>
<td>17.3±1.6</td>
<td>29.1±4.6</td>
<td>22.7±1.7</td>
<td>33.8±5.1</td>
</tr>
<tr>
<td>Junclets</td>
<td>9.2±0.9</td>
<td>17.0±3.1</td>
<td>16.6±1.3</td>
<td>25.3±3.2</td>
</tr>
<tr>
<td>Strokelets</td>
<td>9.7±1.2</td>
<td>17.6±2.6</td>
<td>13.3±1.2</td>
<td>21.9±2.5</td>
</tr>
<tr>
<td>H₂OS&lt;sub&gt;som&lt;/sub&gt;</td>
<td>13.6±1.8</td>
<td>21.6±3.1</td>
<td>18.6±1.7</td>
<td>29.2±3.7</td>
</tr>
<tr>
<td>H₂OS&lt;sub&gt;mlsom&lt;/sub&gt;</td>
<td>6.6±0.7</td>
<td>11.6±2.2</td>
<td>11.7±1.5</td>
<td>18.7±2.8</td>
</tr>
</tbody>
</table>

Table 6.10: The MAEs performance of the dating with different quality configurations.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Train: high Test: low</th>
<th>Train: low Test: high</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge</td>
<td>26.8</td>
<td>15.7</td>
<td>11.1</td>
</tr>
<tr>
<td>Quill</td>
<td>33.6</td>
<td>26.1</td>
<td>7.5</td>
</tr>
<tr>
<td>Junclets</td>
<td>23.6</td>
<td>13.6</td>
<td>10.0</td>
</tr>
<tr>
<td>Strokelets</td>
<td>21.9</td>
<td>16.4</td>
<td>5.5</td>
</tr>
<tr>
<td>H₂OS&lt;sub&gt;som&lt;/sub&gt;</td>
<td>26.0</td>
<td>26.5</td>
<td>0.5</td>
</tr>
<tr>
<td>H₂OS&lt;sub&gt;mlsom&lt;/sub&gt;</td>
<td>18.3</td>
<td>15.6</td>
<td>2.7</td>
</tr>
</tbody>
</table>

Comparison with other studies

In this section, we conduct experiments of dating using more features, such as the SIFT (Lowe, 2004) extracted based on the coarse word zones, (Ojala et al., 2002) (with 255 patterns without the background), CO<sup>3</sup> (Schomaker and Bulacu, 2004), kCF (combined with k=2,3,4,5) and kSF (combined with k=1,2,3) proposed in Chapter 5. Table 6.11 shows the MAEs of different features. From the table we can see that our proposed method achieves better results than other features. In addition, the H₂OS feature is very efficient to represent the handwritten visual elements, such as the Stroke Shape Elements shown in Fig. 6.8.

6.4.5 Geographical localization by classification

To evaluate the document localization performance, we train four classifiers for the four cities. Fig. 6.12 and Fig. 6.13 show the precision of the four cities excluding the writer duplicates and including the writer duplicates, respectively. From the two figures we can find that our proposed methods outperform all the other methods and the H₂OS<sub>mlsom</sub> achieves the best performance in the two configurations. The precision of the H₂OS<sub>mlsom</sub> for Leuven is 90.5% when including writer duplicates and is 86.5% when excluding writer duplicates,
Table 6.11: The dating performance with different methods.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Excluding duplicates</th>
<th>Including duplicates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAEs (±)</td>
<td>CS (α=25) (±)</td>
</tr>
<tr>
<td>LBP</td>
<td>39.9±4.9 59.3%±6.3</td>
<td>34.3±8.1 65.2%±7.7</td>
</tr>
<tr>
<td>SIFT</td>
<td>33.9±4.3 69.9%±3.7</td>
<td>23.3±1.2 78.2%±1.1</td>
</tr>
<tr>
<td>Quill</td>
<td>23.7±2.9 80.5%±3.1</td>
<td>12.1±1.0 89.5%±1.3</td>
</tr>
<tr>
<td>Hinge</td>
<td>22.1±1.9 80.6%±3.1</td>
<td>12.2±0.9 89.6%±1.3</td>
</tr>
<tr>
<td>Junclets</td>
<td>21.7±3.7 79.4%±4.4</td>
<td>12.4±0.7 88.4%±1.4</td>
</tr>
<tr>
<td>CO³</td>
<td>20.3±2.9 82.1%±3.2</td>
<td>11.5±0.8 89.5%±1.5</td>
</tr>
<tr>
<td>Strokelets</td>
<td>19.4±2.7 83.1%±2.6</td>
<td>11.4±0.9 89.4%±1.7</td>
</tr>
<tr>
<td>kCF</td>
<td>19.2±3.5 85.8%±2.8</td>
<td>10.8±0.9 90.8%±1.1</td>
</tr>
<tr>
<td>kSF</td>
<td>17.4±1.9 86.8%±2.0</td>
<td>9.9±0.6 91.8%±1.5</td>
</tr>
<tr>
<td>H₂OSعظم</td>
<td>25.2±3.2 76.4%±3.7</td>
<td>15.8±1.8 85.4%±1.6</td>
</tr>
<tr>
<td>H₂OSmlsom</td>
<td>15.9±2.9 85.4%±3.7</td>
<td>9.1±0.8 91.4%±1.5</td>
</tr>
</tbody>
</table>

Table 6.12: The accuracy (%) of the localization confusion matrix among the four cities using the H₂OSmlsom method when including writer duplicates.

<table>
<thead>
<tr>
<th>City</th>
<th>Arnhem</th>
<th>Leiden</th>
<th>Leuven</th>
<th>Groningen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arnhem</td>
<td>90.9±2.1</td>
<td>7.0±2.0</td>
<td>0.5±0.6</td>
<td>1.6±1.1</td>
</tr>
<tr>
<td>Leiden</td>
<td>2.8±0.9</td>
<td>95.8±0.9</td>
<td>0.6±0.4</td>
<td>0.8±0.4</td>
</tr>
<tr>
<td>Leuven</td>
<td>0.1±0.4</td>
<td>8.9±3.7</td>
<td>90.2±3.8</td>
<td>0.8±0.9</td>
</tr>
<tr>
<td>Groningen</td>
<td>0.6±0.4</td>
<td>0.8±0.6</td>
<td>0.3±0.3</td>
<td>98.3±0.7</td>
</tr>
</tbody>
</table>

which are explicitly higher than other methods. Table 6.12 gives the confusion matrix of the localization of the proposed H₂OSmlsom and the average precision for geographical localization is 93.8%.

### 6.4.6 Results of learning from neighboring neurons

In this section, we evaluate the performance of the learning from neighbors for dating and localization excluding training samples from the target year and city. In this condition, the intrinsic interpolation by the Kohonen map should allow for estimating the target year and city. We leave each combination of labels (year and city) out and train the MLSOM using the rest documents. After that, the documents with the leave-out labels are used to test the trained MLSOM neural network. Table 6.13 shows the MAEs of the results compared to other studies and the CS score is shown in Fig. 6.14. From the results we can observe that due to this local mutilation of the Kohonen map, the MAE approximately equal to 39 years. This is quite natural because the corresponding neurons of the missed labels now contain the general or average information of their neighbors. Although missing the target year
6.4. Experiments

Figure 6.12: The localization precision with different methods in the four cities when excluding writer duplicates.

Table 6.13: The MAEs and CSs(\(\alpha = 25\)) when learning from neighboring years and cities excluding all samples from the target year and city from the training set.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>CS((\alpha=25))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>85.3±58.5</td>
<td>25.7%</td>
</tr>
<tr>
<td>Monk</td>
<td>36.0±20.6</td>
<td>-</td>
</tr>
<tr>
<td>Study [He et al., 2014]</td>
<td>35.4</td>
<td>63.5%</td>
</tr>
<tr>
<td>Learning from neighbors</td>
<td>38.9</td>
<td>63.5%</td>
</tr>
</tbody>
</table>

and city training data, the MLSOM can still learn the information from their neighbors and 14.0% documents are correctly dated (see Fig. 6.14 when \(\alpha = 0\)). For error level \(\alpha=25\) and higher, the CS scores are almost the same as the method in [He et al., 2014]. For historical document localization, 70% documents are localized correctly, compared to 83.8%, the best performance in Table 6.2.
6.4.7 Discussion

The proposed MLSOM neural network contains the year and city label information and can be directly used for predicting the labels of the query document by voting. The results of voting method outperforms the existing systems. If no training data set is available for certain labels, the corresponding neurons learn the average information from their neighbors, with some degradation, but still comparable to our previous results.

In addition, our proposed MLSOM can be considered as a cluster method which contains more discriminative information than traditional cluster methods. The unsupervised cluster methods (such as regular SOM or k-means) discard the subtle difference among labels and are less discriminative in contrast to the proposed MLSOM method. The performance of dating and localization on the MPS data set based on representations using the MLSOM with the classification method achieves state-of-the-art results.
6.5 Conclusion

In this chapter, we studied the problem of historical document dating and localization using our new MPS data set. In order to extract the visual elements in historical documents, we developed the H$_2$OS feature which is a scale-invariant descriptor. We then proposed the Multiple-Label guided MLSOM method to align the visual elements in multiple label space. Our proposed MLSOM can be used for predicting labels directly, or can be used as zero-shot learning by learning from neighbors or can be used as a cluster method. The experimental results in relation to the MPS data set clearly show the efficacy of the proposed method. The best MAE on the MPS data set was 15.9 years when excluding writer duplicates and 9.1 years when keeping writer duplicates in the reference set.

Figure 6.14: Cumulative Scores of $p(\text{MAE} \leq \alpha)$ on the error levels ($\alpha$) from 0 to 100 years when learning from neighboring years and cities excluding all samples from the target year and city from the training set.
Part III

Critical Comparisons