

DateFinder: detecting date regions on handwritten document images based on positional expectancy

Zhenwei Shi

May 2016

Supervisors:

Prof. Lambert Schomaker

Dr. Marco Wiering

Advisor:

Sheng He

Master Thesis

Artificial Intelligence

University of Groningen, the Netherlands



university of
 groningen

faculty of mathematics and
 natural sciences

artificial intelligence

Abstract

Whereas Optical Character Recognition (OCR) technology is used for many documents, such as check, passport, bank statement and receipt, there is a showing interest on modelling of occurrence and location of visual items. However, this source of information attracts much less attention in general OCR. For instance, there are many specific visual items (e.g., dates, writer signatures, calligraphy, author markings, schematic drawings, glyphs and even graffiti) that can be used to explain the underlying meaning and origin of documents. Among the aforementioned visual items, dates play a very important role in many documents (e.g., bank cheques, letters, postal mails, bills and diaries), which can provide time-related clues for readers. Also, the date is central to many administrative applications such as document indexing, translation and retrieval.

In this thesis, we propose a method called DateFinder for detecting date regions on handwritten document images based on a four-step processing sequence. Firstly, we perform pre-processing operations on original scanned images, which aim to extract appropriate proposed date blocks. Secondly, a positional expectancy model is used for ‘date’ text blocks to measure how much an unknown region is similar to a date region based on its position. Thirdly, feature representation and classification techniques are used to extract features from an extracted block and compute the probability this block is a date region. Finally, we combine the scores of the positional expectancy model and classification to determine whether an extracted block is a date region. In the experiments, we have obtained encouraging results for detecting date regions in our dataset. However, there are still ample opportunities to improve the proposed DateFinder method, which can be considered for future work.

Acknowledgements

First, I would like to thank my first supervisor Prof. Lambert Schomaker for his continuous advice and guidance, which made me acquire good comprehension of my master project. He allowed the research to be my own work consistently, but guided me in the right direction.

Second, I heartily acknowledge my second supervisor Dr. Marco Wiering, who always encouraged me to continue my research.

Third, I want to appreciate the PhD student Sheng He for his patience, guidance and assistance throughout the whole project. Without his great support and advices, there would not be a solid basis of my project.

Finally, I must express my sincere gratitude to my dear parents and friends for their consistent help during the period of researching and writing this thesis. This achievement would not have been possible without them.

Thank you very much.

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Research Question	3
1.2 Related Work	4
1.2.1 Detection of Numerical and Alpha-numerical Fields	4
1.2.2 Detection and Recognition of Date Patterns	5
1.2.3 Properties of Specific Datasets	5
1.3 Proposed Approach	6
1.4 Outline	7
2 Technical Background	9
2.1 Visual Attention Model	9
2.2 Histogram of Oriented Gradients	11
2.3 Classification by Support Vector Machines	12
2.3.1 Theory	12
2.3.2 Non-linearity	15
2.4 Convolutional Neural Networks	16
3 Methods	19
3.1 Pre-processing	20
3.1.1 Text Line Segmentation	20
3.1.2 Binarization	21
3.1.3 Morphological Closing Operation	22
3.1.4 Computation of Connected Components	23
3.1.5 Extraction of Proposed Date Blocks	24
3.2 Positional Expectancy Model	24
3.3 Feature Representation and Classification	28
3.3.1 SVM-Based Classifier	28
3.3.2 ConvNets Model	29
3.4 Final Decision	30
4 Experiments and Results	31
4.1 Dataset	31

4.1.1	Data Labelling	31
4.1.1.1	Date Information	33
4.1.1.2	Visually Salient Items	33
4.1.1.3	Data Storage	35
4.1.2	Training, Validation and Test Datasets	35
4.2	Experiments and Results	36
4.2.1	Experiment 1	36
4.2.2	Experiment 2	37
4.2.2.1	Test of SVM-Based Classifier	37
4.2.2.2	Test of ConvNets Model	38
4.2.3	Experiment 3	40
4.3	Final Evaluation of DateFinder Method in Free Search	40
5	Discussion	43
5.1	Dataset Challenges	43
5.1.1	Different Date Formats	43
5.1.2	Undetermined Position of Date Regions	44
5.1.3	Low Quality of Images	44
5.1.4	Touching Characters	45
5.1.5	Mis-classification among Date, Numeral and Letter	45
5.2	ConvNets Model and SVM-based Classifier	46
6	Conclusion and Future Work	47
6.1	Conclusion	47
6.2	Future Work	48
6.2.1	DateFinder Optimization	48
6.2.2	Additional Model for Detecting Date Regions	48
6.2.3	DateFinder Extension	49
	Bibliography	51

Chapter 1

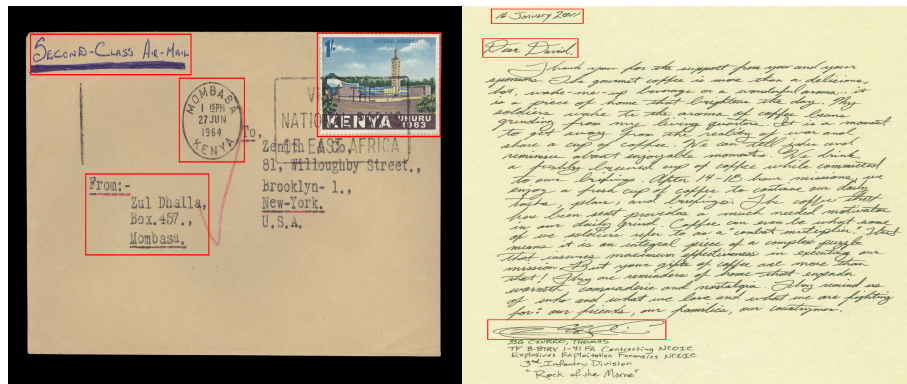
Introduction

Nowadays, a massive number of handwritten documents have been scanned into images and stored in database depositories. Due to the advantages including storing easily, managing conveniently and accessing from anywhere at any time, digitally scanned documents are widely used in a number of applications.

One of the most widely used techniques for document analysis is Optical Character Recognition (OCR), which can convert scanned document images into editable and searchable documents. However, the OCR systems [1] typically are not powerful enough to deal with handwritten manuscripts. The possible reasons are connected cursive texts, various types of noise (e.g., small speckles and lines), character types and handwritten items, which make it difficult to recognize characters in handwritten documents. Thus there is a need to design a system that can automatically index and retrieve visual items on handwritten document images.

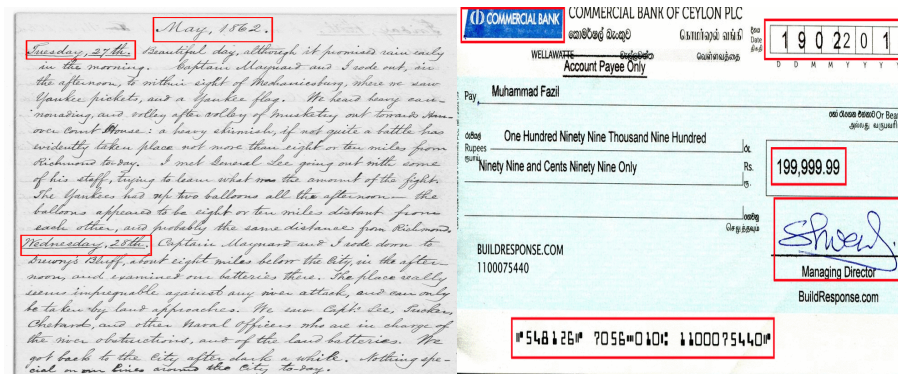
As an instance of this kind of system, a search engine called Monk system [2] has been proposed for handwritten text indexing and retrieval of historical handwritten document images, which are hard to process through traditional Optical Character Recognition (OCR) methods. The Monk system consists of two main parts: 1) scanned document images with annotation (i.e., datasets) and 2) a collection of approaches and algorithms for recognition, retrieval and searching of handwritten texts. More details about the Monk system can be found in [2–4].

To design a system for automatically detecting visual items on historical handwritten document images, we first analyse different types of visual items in a handwritten document. In general, visual items can be classified into two classes: 1) text paragraphs and 2) specific visual items (e.g., titles, dates, writer signatures, author markings, schematic drawings and even graffiti):



(A) Postal Mail.

(B) Letter.



(C) Diary.

(D) Bank Cheque.

FIGURE 1.1: Examples of visually salient items in handwritten documents. Images taken from Google Images.

- Text paragraphs are usually the primary parts of a document, which consist of concrete contents (e.g., stories, incidents or phenomena) that authors want to express to readers.
- Specific visual items also play a very important role in documents, which can help readers understand the underlying meaning and provenance of documents. Generally, the properties (e.g., appearance, shape, spatial position and color) of these visual items are specific, which make them stand out from their contexts. These specific items can be considered as salient information that is derived from the study of the human visual system. When we are scanning a document, the salient information can attract the human attention immediately without any prior knowledge [5]. Fig.1.1 shows some examples of visually salient items that stand out from their contexts in handwritten documents, such as postal mail, letter, diary and bank cheque.

In this thesis, we only focus on dates in handwritten documents, which can provide clues of time-related information for readers. The date is central to many administrative

applications including document indexing, translation and retrieval. For instance, the date allows users to reconstruct chronology in historical diary datasets.

1.1 Research Question

Throughout this project, the ultimate objective can be summarized as: ***“How to detect date regions on handwritten document images?”***. The proposed approach for detecting date regions in this thesis is performed in text-block level, hence the research question can be re-summarized as ***“How to classify a handwritten text block as being a date?”***.

For this research question, we first analyse the properties of date patterns in both general and specific datasets:

- Dates usually have regular expressions or widely used formats in general datasets, which can be considered as a common property of dates. For instance, the complete format of a date usually consists of ‘year’, ‘month’ and ‘day’.
- Dates might have special properties (e.g., position, color and appearance) in specific datasets, which can be used to distinguish date patterns from other visual items in handwritten documents.

The dataset used in this thesis is called Diary of Leo Polak (DoLP) dataset, which consists of 53 volumes of diaries written in Dutch by the Dutch philosopher Leo Polak ¹ from 1905 to 1941. Fig.1.2 shows several examples of dates in the DoLP dataset. These examples present three significant properties:

- Various date patterns such as only word (‘April’), only numeral (‘4’), only abbreviation (‘Okt’) or their combination (‘21,Okt,1901’) exist in the DoLP dataset.
- There is no obvious regular expression of dates that can be applied to represent these dates. The handwritten documents used in this thesis are personal diaries that are considered as informal documents.
- Dates have the property of undetermined position in the DoLP dataset, which might be written anywhere in a page (e.g., top-left or right corners, bottom-left or right corners or in body paragraphs).

Therefore, the goal of this thesis is to propose a robust method, which can detect date regions on handwritten document images that have complex contexts.

¹Leo Polak: https://nl.wikipedia.org/wiki/Leo_Polak

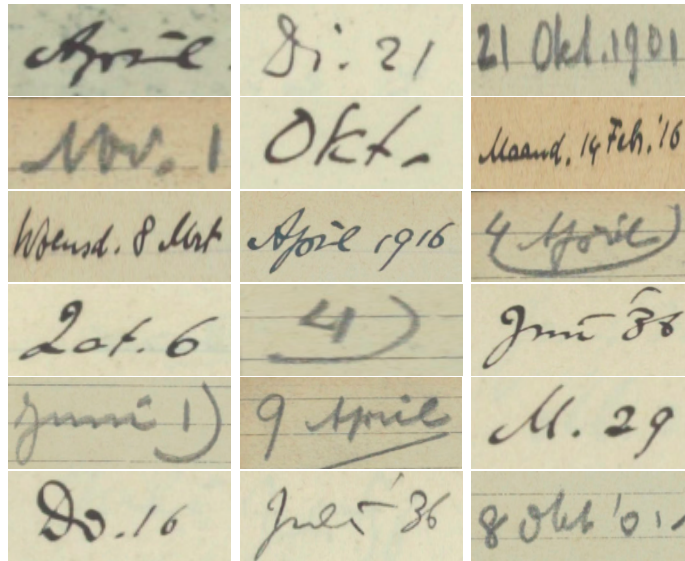


FIGURE 1.2: Different date patterns collected from the DoLP dataset.

1.2 Related Work

In this part, we first describe some published studies on automatically detecting numerical and alpha-numerical fields on handwritten document images in Section 1.2.1. Second, a few studies about detection and recognition of date patterns on specific documents (e.g., bank cheques) are presented in Section 1.2.2. Finally, Section 1.2.3 discusses the properties of the datasets used in the studies described in Section 1.2.2.

1.2.1 Detection of Numerical and Alpha-numerical Fields

A Hidden Markov Model (HMM) based system has been proposed by Thomas et al [6], which can extract fields of alpha-numerical sequences from handwritten document images. Compared to other existing methods, the method in [6] uses a global handwriting line model, which can describe two types of information: relevant and irrelevant information. A shallow parsing model was designed to represent this information. In this way, the relevant information can be detected from handwritten document images through the shallow parsing model.

Koch et al. [7] also proposed an approach based on the HMM, which can automatically detect numerical sequences on handwritten document images. The approach is based on a syntactic analyzer over individual text lines.

Chatelain et al [8] proposed a method to extract numerical sequence fields (e.g, ZIP code and phone number) from incoming mails. First, the segmentation-driven recognition is

performed to detect isolated and touching digits. Second, the sequences satisfying a specific syntax are selected through syntactical analysis that is performed on individual text lines.

The aforementioned studies only deal with automatically detecting fields of numerical or alpha-numeric sequences rather than date regions on handwritten document images. In the following part, a few studies on detection and recognition of date patterns in specific handwritten documents (e.g., bank cheques) are described.

1.2.2 Detection and Recognition of Date Patterns

A method has been proposed to automatically detect and recognize date patterns that are written on bank cheques by Suen et al. [9]. First, date images are extracted by using separators. Second, the fields of Year, Month and Day are localized according to shape and spatial features. Third, the regions of the numeric and non-numeric month are recognized by using two recognizers, which are designed for connected digits and cursive words respectively. Finally, a parsing model is applied to select valid candidates and reject invalid ones.

Recently, Mandal et al. [10] proposed a four-step classification-based method for extracting date regions in handwritten documents. First, words in each text line are classified into ‘month’ and ‘non-month’ category based on word-level features. Second, individual ‘digit’, ‘punctuation’ or ‘alphabet’ components are found by using component-level features. Third, candidate lines are detected through a voting method. Finally, date patterns are extracted by using regular expressions of dates.

In addition, Mandal et al. [11] proposed a framework for automatic extraction of date regions on handwritten document images, where sliding window-based Local Gradient Histogram (LGH) features and a character-level HMM-based method are used for segmentation and recognition respectively. First, individual date patterns are segmented into components, which are labelled as month-word, numeral, punctuation and contraction. Then numeric and semi-numeric date regions are extracted based on regular expressions of dates.

1.2.3 Properties of Specific Datasets

The datasets used for detecting and recognizing date patterns above present a few properties: 1) specific types of documents; 2) particular syntax or regular expressions of dates; 3) target surrounded by very different neighbours.

- Most of handwritten materials (e.g., bank cheques) used in their studies are generally composed of a small number of machine printed and handwritten texts. Thus, it is easier to detect handwritten dates in these specific documents (e.g., datasets used in [9, 12]) than in the documents that have many handwritten texts and complicated contexts.
- The dates are generally in a complete form in these documents, which are composed by ‘Year’, ‘Month’, ‘Day’ and separator components (e.g., ‘,’ and ‘/’). Thus, it is reasonable to classify a handwritten text pattern into date or non-date class by using a sliding window method and extracting date patterns through a regular expression of a date [10, 11]. However, a date can have various formats in real life, which might make the searching strategy based on regular expressions of dates failing. For instance, although the documents in our dataset are handwritten diaries written by one author, the dates have different formats including word, numeral, abbreviation and their combinations. The author can use a single numeral ‘9’, a week ‘Zondag’ or an abbreviation ‘Zond’ to express the exact date ‘9, Zondag, October, 1910’.
- The desired target fields are surrounded by very different patterns [6–9, 12]. For instance, dates are different from common texts, which makes it easier to distinguish them based on the shape and spatial features. However, if date patterns are very close to numerals in a page, it can lead to mis-classification between numerals and dates.

Thus, there is a need to propose a robust system that can solve the problems including various date formats, irregular expressions of dates and complicated contexts for detecting date regions on handwritten document images.

1.3 Proposed Approach

We propose a method called DateFinder for detecting date regions on handwritten document images, whose flow diagram is shown in Fig.1.3. The proposed DateFinder method follows a four-stage processing sequence, which is described as follows:

In the pre-processing stage, the operations include text line segmentation, binarization, morphological closing, computation of connected components and noise reduction. The operations are performed on scanned images, which aim to extract appropriate proposed date blocks. In the second stage, we propose a statistical learning-based positional expectancy model, which is designed by analysing the positions of date patterns collected

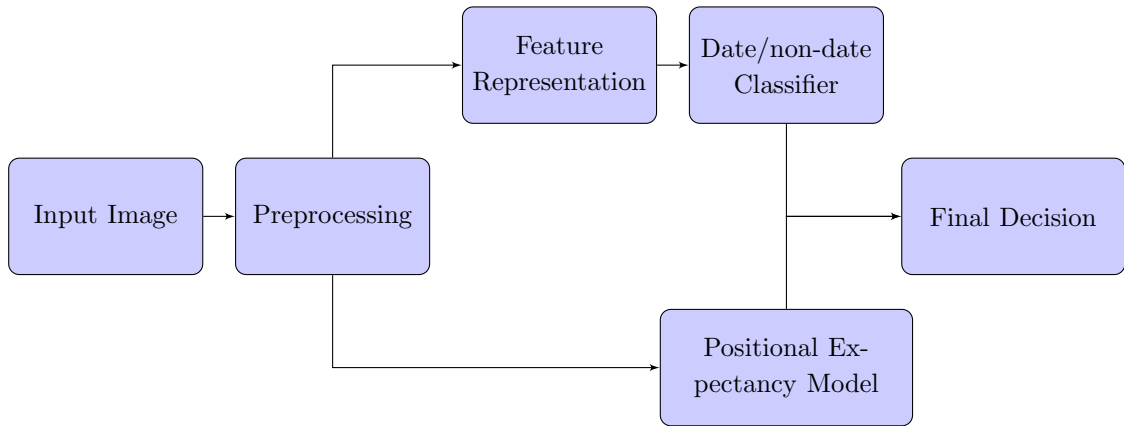


FIGURE 1.3: The flow diagram of the proposed DateFinder method.

in the DoLP dataset. The proposed positional expectancy model can measure how much an extracted block is similar to a date region based on its position. Third, features are extracted from an extracted block by using feature representation techniques and then fed into a date /non-date classifier. Finally, the positional expectancy model and classification scores are combined to make a final decision, which can determine whether an extracted region is a date.

Since the proposed approach cannot extract the exact position (i.e., a narrow contour) of a date, we use a bounding box to indicate a date field. A successful date region detection will be defined as the system extracting a block that has an overlap equal or more than 50 percent with the bounding box of a date and the ground truth.

1.4 Outline

In Chapter 2, we describe the background of the techniques explored in this thesis in detail. First, it starts with a description of the visual attention model [13]. Second, the process of the HoG [14] feature extraction is presented. Third, the details of how SVMs [15] work both in linear and non-linear modes are described. Finally, we describe the convolutional neural networks (ConvNets) [16], which is a robust approach for image classification. Each individual processing stage in the proposed DateFinder method is described in Chapter 3. The details of the dataset, experiments and results are presented in Chapter 4. Chapter 5 gives a discussion for this thesis. Finally, a conclusion is drawn in Chapter 6, which also includes some future work for further improvements.

Chapter 2

Technical Background

In this chapter, we start with the theoretical background of the visual attention model [13] in Section 2.1, which is used to explain how the human visual system works for detecting targets among a number of received data. Second, the histogram of orientated gradients (HoG) [14] is described in Section 2.2, which is widely used for object detection. Third, Section 2.3 presents the support vector machine (SVM) [15], which has many advantages (e.g., easy implementation, simple structure and less computation) for pattern classification. Finally, Section 2.4 describes the Convolutional neural networks (ConvNets) [16], which can combine feature extraction and classification into an end to end framework.

2.1 Visual Attention Model

Visual attention is a capability of the human visual system, which has been studied in many fields (e.g., cognitive psychology [17], neuroscience [18] and computer vision [19, 20]). As described in [21], human eyes can receive a large stream of visual data each second, which makes it hard to process these data in real time. For instance, a target detection task requires humans to find relevant targets from numerous received data. However, it is effortless for humans to detect targets in a natural scene, which attributes to the selective mechanism that can help humans pay attention to the useful data and ignore the irrelevant data.

Over the past 25 years, modelling the human visual attention has become a hot topic, especially the salience-driven and task-driven attention [22]. Following early attention models [5] and cognitive theories [23], a number of visual attention models have been proposed and applied for target detection in images and videos.

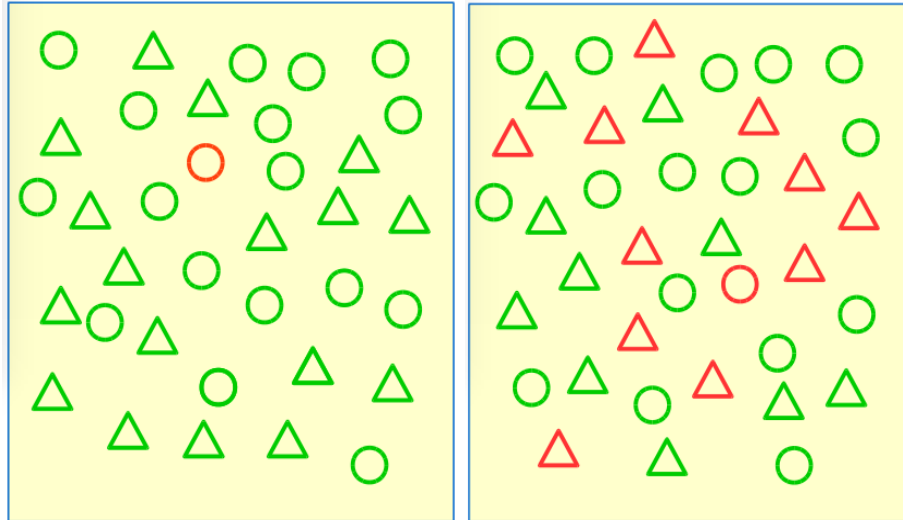


FIGURE 2.1: Examples of the salience-driven and task-driven mechanisms. The human attention is almost immediately attracted to the visually salient object (red circle) in the left image based on the salience-driven attention. On the other hand, the task-driven attention refers to answering the question of “Is there a red circle?” in the right image. In order to answer this question, a sequential procedure needs to be performed.

Visual attention models can be classified into two categories: 1) bottom-up and 2) top-down processing. First, the bottom-up processing is also known as the salience-driven attention, which is fast, salience-driven, involuntary and task-independent [22]. The visual items that attract the human attention in the salience-driven manner must be sufficiently different from their contexts. Hence, the bottom-up attention is mainly based on the properties of visual items. Second, the top-down processing (i.e., goal-driven attention) is slow and driven by goals or intentions, which mainly relies on prior knowledge and expectancy [18]. Fig.2.1 shows two examples, which are used to explain these two types of mechanisms. On the one hand, the human attention is drawn to the red circle directly within the left image mainly based on the color property. On the other hand, humans are required to answer the question: “Is there a red circle within the image?” based on the task-driven mechanism in the right image.

Unfortunately, dates do not constitute a very salient pattern in a open context. The question is whether dates are haphazardly distributed over a page. Fig.2.2 shows an example image in the DoLP dataset, where dates were manually labelled by red rectangles. It appears that the positions of dates may follow a particular distribution, which makes date regions stand out from their contexts. Therefore, we propose a positional expectancy model that can detect date regions on handwritten document images.

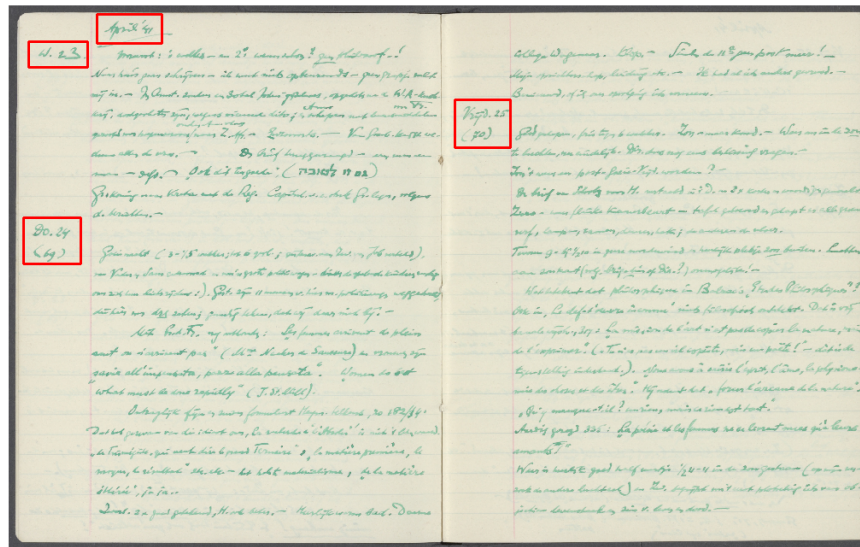


FIGURE 2.2: An example for date regions (i.e., red rectangles) separated from other visual items in the DoLP dataset.

2.2 Histogram of Oriented Gradients

The histogram of oriented gradients (HoG) [14] is a popular feature descriptor, which has been widely applied in computer vision and image processing fields. The idea behind the HoG feature is that the appearance and shape of a local object can be described by the distribution of local intensity gradients within the bounding box of the object.

The implementation of extracting the HoG feature of an object follows three steps. First, we need to compute the gradient of each pixel within a small zone called cell. Second, histograms of gradient directions are created over each cell. Each histogram has several bins, which are equally spaced over $0^{\circ} - 180^{\circ}$ or $0^{\circ} - 360^{\circ}$ based on the usage of signed or unsigned gradient values. Finally, the concatenation of these 1D histograms forms the feature descriptor.

For better accuracy against the changes of illumination and shadowing, we contrast-normalize the local histograms before using them. In addition, a measure of the intensity within a larger region (block) is calculated and all cells are normalized within a block. The normalized histogram is referred to the histogram of oriented gradients (HoG) descriptor [14].

Feature extraction plays a very important role in many object detection and recognition systems. In this thesis, the Rectangle Histogram Oriented Gradient (R-HoG) is used for detecting date regions on handwritten document images. Fig.2.3 shows some examples of the R-HoG features extracted from handwritten text images in the DoLP dataset.

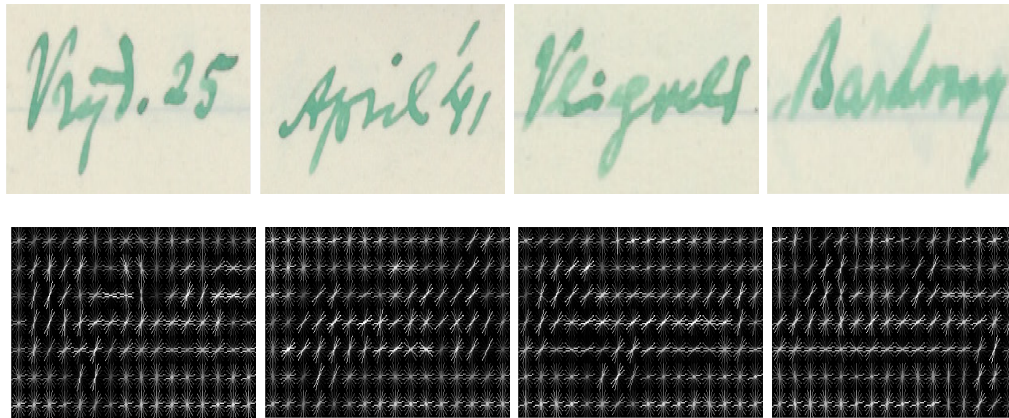


FIGURE 2.3: Visualization of the HoG [14] feature.

2.3 Classification by Support Vector Machines

The original support vector machine (SVM) algorithm is derived from the Vapnik-Chervonenkis theory developed during 1960 to 1990 by Vladimir Vapnik and Alexey Chervonenkis. Cortes and Vapnik introduced the current form of SVMs [15], which can be considered as models trained by using appropriate supervised learning algorithms for data analysis and pattern classification. The SVMs are usually described in two dimensions, however many cases involve higher dimensions in real life. For a convenient explanation, we assume that the separating hyperplanes used to separate data points from different classes are two dimensional lines in the following description.

2.3.1 Theory

The basic goal of the SVM is to find the separating hyperplane in a feature space. As shown in Fig.2.4, the dashed line indicates the separating hyperplane, which can be imagined as the median line of a ‘street’. The solid lines (i.e., two ‘gutters’ of a ‘street’), which are situated on one or more data points belonging to each class, determine the width of the ‘street’. Hence, the optimal separating hyperplane can be defined as the hyperplane that makes the width of the ‘street’ largest. In other words, the essence of the SVM is margin maximization. To find the optimal separating hyperplane, we follow a five-step method, which is described as follows.

In the first step, we need to define the decision rule, which determines the side of the separating hyperplane where an unknown data point lies on. For this purpose, we follow three sub-steps. As shown in Fig.2.4, the positive data points are described by crosses and the negative data points are described by circles. First, we create a vector $\vec{\omega}$, which is perpendicular to the median line of the ‘street’. Second, an unknown data point u and

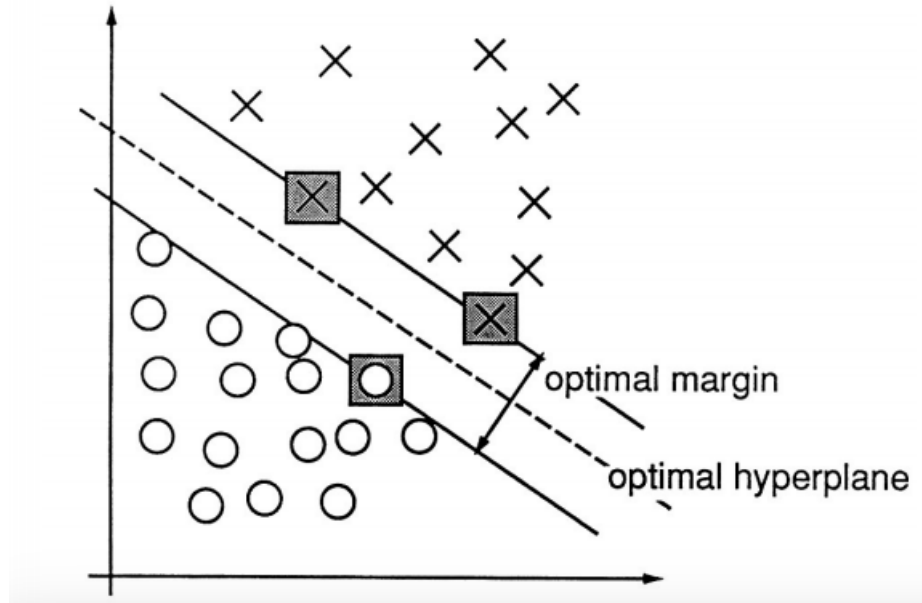


FIGURE 2.4: A 2D binary classification problem. One class is described by crosses and the other one is described by circles. The support vectors are shown by the grey squares, which determine the largest margin between the two classes. The dashed line indicates the optimal hyperplane that best separates the classes from each other. The image is taken from [15]

the origin $(0, 0)$ create a vector \vec{u} . Finally, the side of the hyperplane where u lies on can be predicted by taking the dot product and a constant c : $\vec{\omega} \cdot \vec{u} \geq c$. The equation can be rewritten by applying $b = -c$, which leads to the decision rule described in equation 2.1. When it is true, the unknown data point u is a positive example, otherwise u is a negative example.

$$\vec{\omega} \cdot \vec{u} + b \geq 0. \quad (2.1)$$

As described in equation 2.2, we want to insist that if a data point belongs to the positive class, the result should be equal or greater than 1. The equation 2.3 denotes that the result should be equal or less than -1 , if a data point is from the negative class.

$$\vec{\omega} \cdot \vec{x}_+ + b \geq 1. \quad (2.2)$$

$$\vec{\omega} \cdot \vec{x}_- + b \leq -1. \quad (2.3)$$

Secondly, a new variable y_i is used to combine equations 2.2 and 2.3 into one equation, where $y_i = +1$ for positive examples and $y_i = -1$ for negative examples. Hence:

$$y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1. \quad (2.4)$$

For all support vectors, equation 2.4 can be simplified even further to form:

$$y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1 = 0. \quad (2.5)$$

Thirdly, we want to give an expression to compute and maximize the width of the margin. Based on equation 2.5, we can compute the width of the margin by taking the difference ($\vec{x}_+ - \vec{x}_-$) between two data points that are located on the ‘gutters’ from different classes. We take the dot product of the difference with the unit vector of $\vec{\omega}$. Hence:

$$|\vec{x}_+ - \vec{x}_-| \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|}. \quad (2.6)$$

Since y_i is +1 for positive examples and -1 for negative examples, equations 2.5 and 2.6 can be combined to form a much simpler expression for the width of the margin:

$$\frac{2}{\|\vec{\omega}\|}. \quad (2.7)$$

Fourthly, the margin equation 2.7 can be maximized by minimizing $\|\vec{\omega}\|$. Also, the maximal margin can be achieved by minimizing $\frac{1}{2}\|\vec{\omega}\|^2$ for later mathematical convenience. To solve this optimization problem, the Lagrange multipliers is used to generate a new minimization function L , which is described by:

$$L = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_i \alpha_i [y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1]. \quad (2.8)$$

The minimization problem can be resolved by computing the derivatives and setting them to 0. For $\vec{\omega}$, we obtain equation 2.9:

$$\frac{\partial L}{\partial \vec{\omega}} = \vec{\omega} - \sum_i \alpha_i y_i \vec{x}_i = 0. \quad (2.9)$$

Hence, $\vec{\omega}$ can be described by:

$$\vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i. \quad (2.10)$$

For constant b , we obtain equation 2.11:

$$\frac{\partial L}{\partial b} = - \sum_i \alpha_i y_i = 0. \quad (2.11)$$

Hence:

$$\sum_i \alpha_i y_i = 0. \quad (2.12)$$

Equation 2.10 is filled in equation 2.8 resulting in:

$$L = \frac{1}{2} \left(\sum_i \alpha_i y_i \vec{x}_i \cdot \sum_j \alpha_j y_j \vec{x}_j \right) - \left(\sum_i \alpha_i y_i \vec{x}_i \cdot \sum_j \alpha_j y_j \vec{x}_j \right) - \sum_i \alpha_i y_i b + \sum_i \alpha_i. \quad (2.13)$$

Since b is a constant, equation 2.13 can be rewritten into a much simpler form:

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_j \alpha_i y_i y_j \vec{x}_i \cdot \vec{x}_j. \quad (2.14)$$

Finally, we conclude that: the maximization problem is only relying on the dot product of the samples $\vec{x}_i \cdot \vec{x}_j$ through equation 2.14,. If equation 2.10 is filled into the decision rule (i.e., equation 2.1) with an unknown sample \vec{u} , we can obtain:

$$g(x) = \sum_i \alpha_i y_i \vec{x}_i \cdot \vec{u} + b \geq 0. \quad (2.15)$$

If equation 2.15 is true, the sample \vec{u} belongs to the positive class, otherwise it belongs to the negative class.

2.3.2 Non-linearity

The original version of the SVM proposed by Vapnik in 1963 can only solve linear problems. However, there are many non-linear problems in practical situations. Fortunately, the SVMs using kernels can be applied for non-linear classification, which makes it much more robust in real life cases. By using the mapping function $\phi(\vec{x})$, the kernels allow classification operations in an implicit feature space that has higher dimensions than the dimensions of datasets. As described above, we want to maximize equation 2.6. Thus, we can wrap the mapping function around the two vectors of the dot product ($\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ and $\phi(\vec{x}_i) \cdot \phi(\vec{u})$), which leads to a kernel function K :

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d. \quad (2.16)$$

One of the most important advantages of the kernel function is that we do not have to resolve the original mapping function $\phi(\vec{x})$, which is known as the kernel trick originally

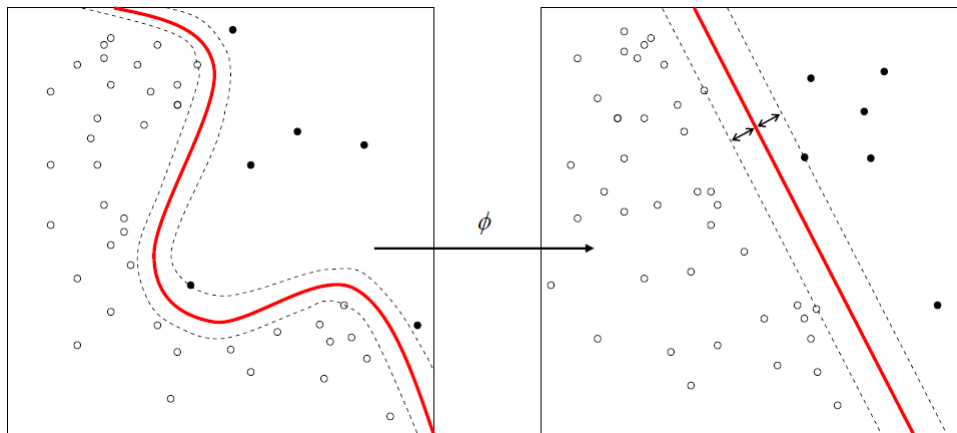


FIGURE 2.5: An example for a kernel operation transforming a non-linear classification problem into a linear classification problem. The image is taken from Wikipedia user Alisneaky, public domain license.

proposed by Aizerman et al. in 1964 [24]. Fig. 2.5 shows a visual description of non-linear SVMs.

For classification problems, four types of kernels can be used in SVMs models: linear, polynomial, radial basis function (RBF) and sigmoid. The one used in this thesis is the the RBF kernel function [25], which is described in equation 2.17.

$$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2), \gamma > 0. \quad (2.17)$$

2.4 Convolutional Neural Networks

Deep learning can be viewed as a family of approaches, which applies deep architectures (i.e., combinations of many layers) to discover and learn features in datasets [26]. When a network has more than one hidden layer, we usually call it a deep neural network. As described in [27], deep neural networks can detect and learn hierarchical features from input data, where higher-level features are computed from lower-level features.

Deep learning techniques are powerful to process the data with complex structures and high dimensions, which have shown outstanding performance in many applications. First, deep learning techniques have achieved the best results in image recognition [16, 28–30] and speech recognition [31–33]. Second, they have shown better performance than many other machine learning methods (e.g., described in [34, 35]). Apart from the aforementioned applications, deep learning techniques have achieved encouraging results in natural language understanding [36], question answering [37] and language translation [38, 39].

Three factors contribute to the improvement of performance:

1. More available training datasets, which consist of a large number of labelled examples.
2. Powerful computation ability of graphics processing units (GPUs), which make large and complicated computation manageable.
3. Better model regularization approaches (e.g., dropout [40]), which can decrease the effect of overfitting.

There are many different variants of deep architectures including Convolutional Neural Networks (ConvNets) [16], Deep Belief Networks [41] and Deep Boltzmann Machines [42]. The one used in this thesis is the ConvNets, which is a feed-forward artificial neural network.

In general, three types of layers are used in a ConvNets architecture: convolutional layer, pooling layer and fully connected layer. As described in [26], the convolutional layers and pooling layers are often applied in the front of the ConvNets architecture, which are derived from the classic concepts of simple and complex cells in the visual neuroscience field [43]. The fully connected layer is similar to the one in regular neural networks. The detailed description of each type of layer is presented as follows:

1. The convolutional layer consists of a set of feature maps, which is used to extract multiple features at each position within an image. Units are arranged in feature maps and linked to units in the previous layer. To achieve this, a filter bank composed by a set of weights is used. Then, the local weighted sum is fed to a non-linear function (e.g., a rectified linear unit (ReLU) [16]).
2. The pooling layer is used to reduce the spatial size of the representation, which can result in reduction of the amount of parameters and computation in a network. A widely used pooling layer is the max-pooling layer, which can compute the maximal value of a local patch in one or more feature maps. In the max-pooling processing, adjacent units can use the maximal value to decrease the dimension and generate an invariance to small shifts and distortions [26].
3. Neurons in fully connected layers have all possible connections to the neurons taken from the previous layer. The activations can be computed through a matrix multiplication followed by a bias offset.

Among many variants of ConvNets architectures, one of the most widely used is the LeNet-5 [44] proposed by LeCun et al. in 1998, which is shown in Fig.2.6. The LeNet-5

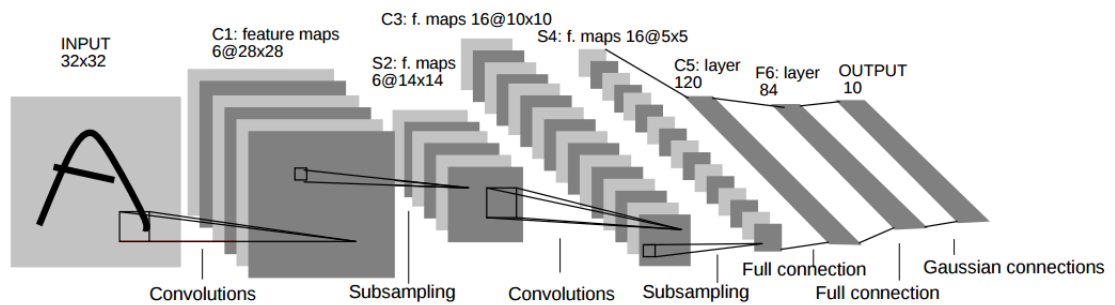


FIGURE 2.6: Architecture of LeNet-5, a Convolutional Neural Network, for digits recognition. Image taken from [44]

architecture has shown excellent performance for handwritten digit classification [44, 45] and face detection [46, 47].

From the introduction of LeNet-5 [44], the ConvNets have had a standard structure, which contains a few convolutional layers that are optionally followed by non-linearity (e.g., rectified linear units (ReLUs)) and pooling layers, then one or more fully-connected layers are used. All the weights in filter banks are learned by using a back-propagation method as the same learning approach as applied in regular neural networks.

For a classification problem, deep neural networks can amplify the features that are useful for classification and suppress the useless features by higher layers of representation [26]. As described in [26], features about the presence or absence of objects' edge information can be detected in the first layer. In the second layer, motifs can be found through specific arrangements of edges. In the third layer, motifs can be gathered into larger parts that are similar to parts of familiar objects. The following layers would detect objects based on the combinations of these parts. Since the features in each layer can be automatically found and learned from input data, it is not necessary to design them by hand, which can be considered as a main advantage of deep neural networks [26].

Chapter 3

Methods

The proposed DateFinder method for detecting date regions on handwritten document images follows four steps: 1) pre-processing, 2) computation of positional expectancy, 3) feature extraction and classification and 4) final decision. Fig.3.1 shows the pipeline of the proposed method, where each step is represented by one type of color.

Firstly, the pre-processing is performed for extracting appropriate proposed date blocks, which is described in Section 3.1. Secondly, the positional expectancy model is detailed in Section 3.2, which is used to measure how much an extracted block is similar to a date region based on its position. Thirdly, we propose two methods for date feature extraction and date/non-date pattern classification: 1) an SVM-based classifier and 2) a ConvNets model, which are described in Section 3.3. Finally, Section 3.4 presents the final decision algorithm for detecting date regions on handwritten document images.

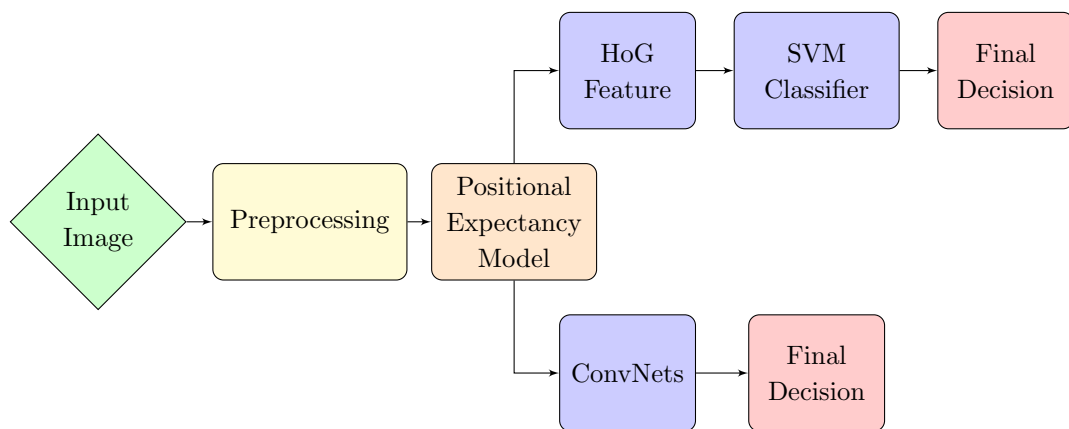


FIGURE 3.1: Architecture of the proposed DateFinder method. The upper pipeline shows the method by using the HoG [14] feature descriptor and the SVM [15] classifier. The bottom pipeline shows the ConvNets [16] method.

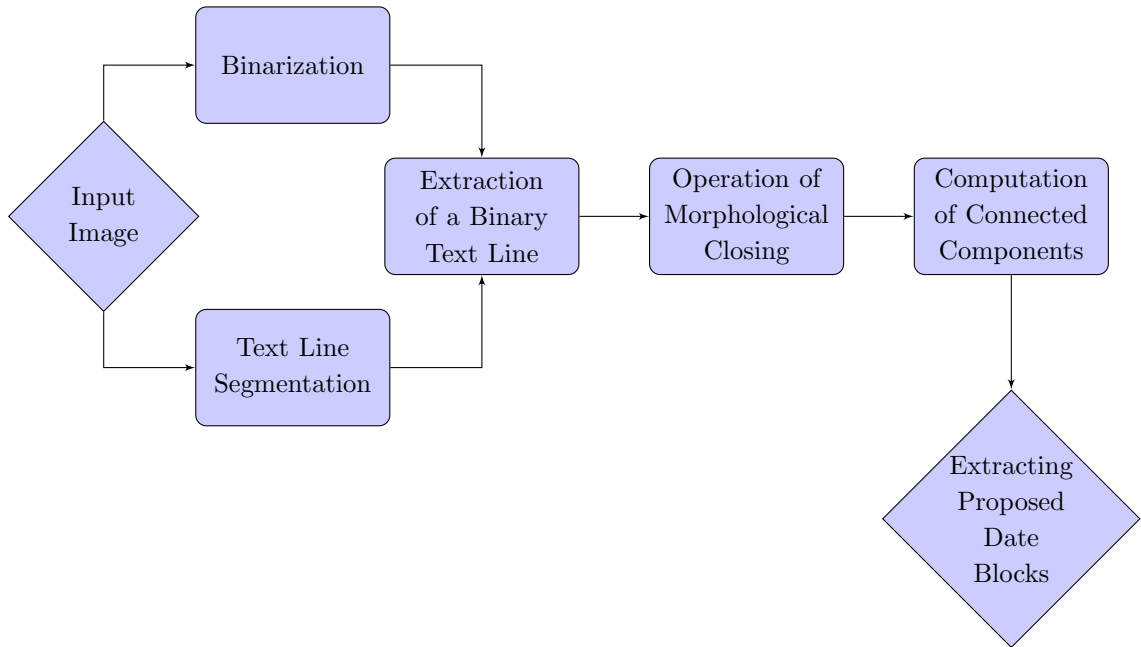


FIGURE 3.2: The flow chart of each operation in the pre-processing stage. Firstly, the input image is processed through a single text line segmentation algorithm and a binarization method at the same time. Secondly, the operation of morphological closing is performed on binary images to reconstruct the shape of characters. Thirdly, we compute the connected components twice to reduce noise and determine the position of text blocks. Finally, the proposed date blocks are extracted and reserved, which can be used for the following stages.

3.1 Pre-processing

We first need to perform pre-processing on original scanned images to extract the proposed date blocks for the following stages. The main reasons for this are twofold. Firstly, the proposed DateFinder method is based on classifying an extracted block into date or non-date classes, hence text blocks should be extracted from a scanned image for later classification. Secondly, the scanned images are in color-scale in our dataset, which results in failure of directly computing connected components. Hence we need to convert the color-scale images into binary images, using image binarization. Fig.3.2 shows the pre-processing pipeline, where the operations include text line segmentation, binarization, morphological closing, computation of connected components, noise reduction and extracting proposed date blocks.

3.1.1 Text Line Segmentation

First, the operation of text line segmentation is performed on original scanned images by using a robust algorithm called seam carving [48]. Text line segmentation is an important pre-processing operation, which has been applied in many applications (e.g.,

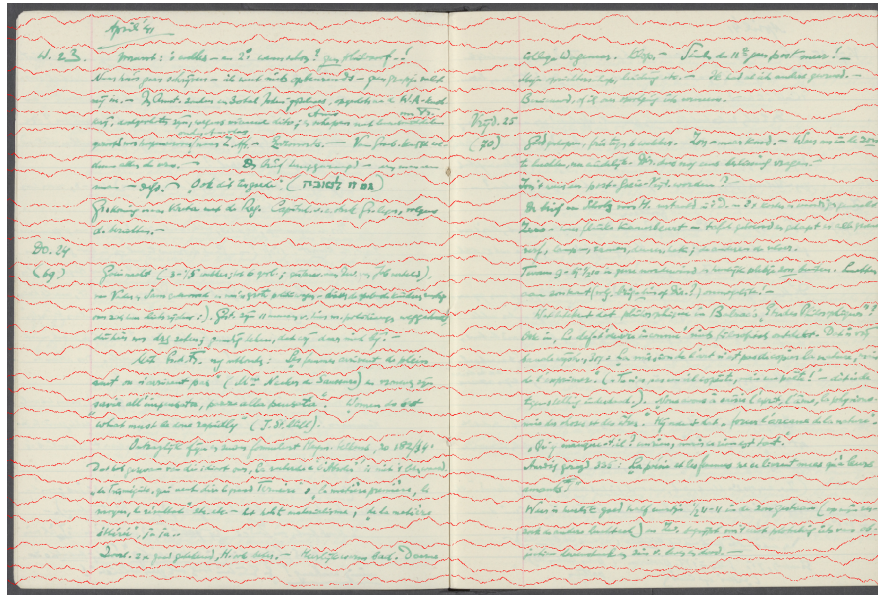


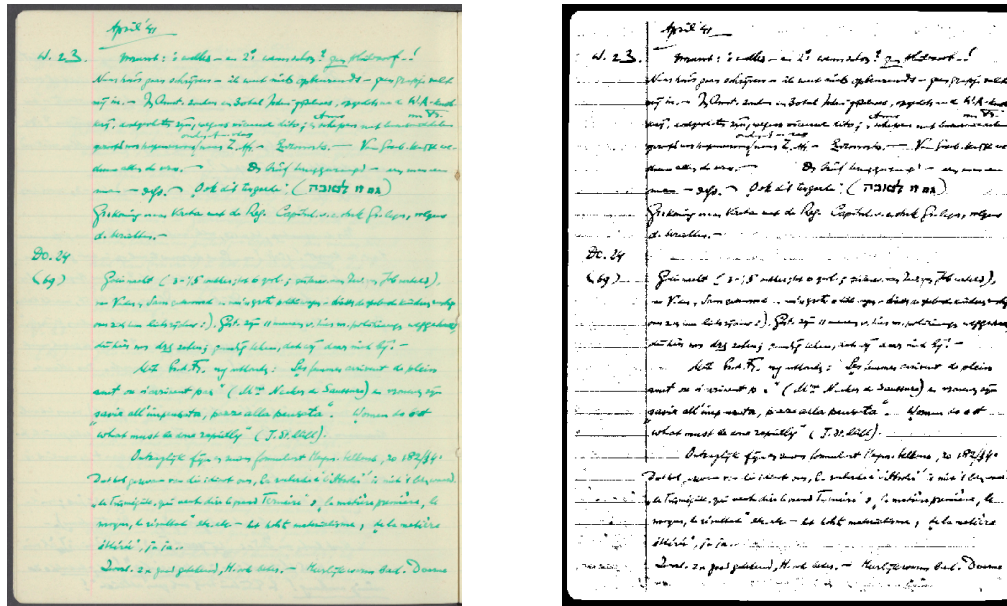
FIGURE 3.3: A color-scale handwritten document image is divided into individual lines by using the seam carving algorithm [48], where the separating seams are represented by red lines.

document structure extraction, keywords searching and handwriting recognition). However, there is a common problem caused by text line segmentation performed on binary images: the binarization operation might cause severe information loss on low quality scanned images, which can result in poor performance. In contrast to general text line segmentation techniques, the seam carving algorithm [48] allows extraction of individual text lines on color or gray-scale scanned images without prior binarization, which can solve the problem of information loss caused by the operation of image binarization.

The idea behind the seam carving algorithm is to compute the separating seams between two consecutive text lines without cutting through line components. To achieve this goal, we need to compute the energy of seams within a document image, where high-energy zones denote texts and low-energy zones denote blank paper or document background. Fig.3.3 shows an example of a handwritten document image in the DoLP dataset, which has been segmented into several individual text lines by using the seam carving algorithm [48].

3.1.2 Binarization

Secondly, since the operation of computing connected components must be performed on binary images, a fast and efficient thresholding approach called Otsu's method [49] is used to automatically convert original images to binary images in this thesis. Otsu's method has been widely applied in computer vision and image processing fields. More



(A) Original image

(B) Corresponding binary image

FIGURE 3.4: An scanned image and its corresponding binary image computed by using Otsu's method [49]

details of Otsu's method can be found in [50]. Fig.3.4 shows an example image and its corresponding binary image computed by using Otsu's method.

3.1.3 Morphological Closing Operation

Thirdly, the morphological closing operation (i.e., dilation followed by erosion) [51] is performed on binary images obtained from the previous stage, which is used to reconstruct the shape of characters.

The morphological operation is based on a structuring element (S.E.), which can be simply considered as a mask determining arbitrary surrounding structure. On the one hand, the exact operation can maintain the background fields, whose shape is similar to the applied S.E. or which thoroughly contain this S.E.. On the other hand, the used S.E. can eliminate all other zones of background pixels. The idea behind the closing operator is to extend the boundaries of foreground fields and contract the background holes within an image, which leads to less destruction of original boundary shapes.

As described above, the operation of image binarization can cause information loss at different levels, which may result in a lot of noise (e.g., small points) and inaccurate computation of connected components. Hence, it is essential to perform the operation of morphological closing before computing connected components [51].

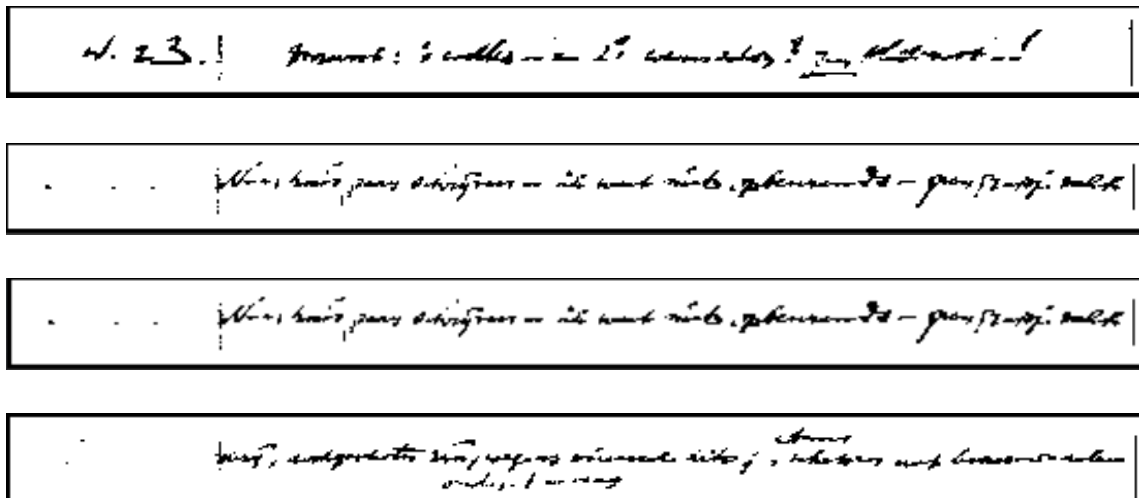


FIGURE 3.5: Examples of individual and binary text lines without removing noise.

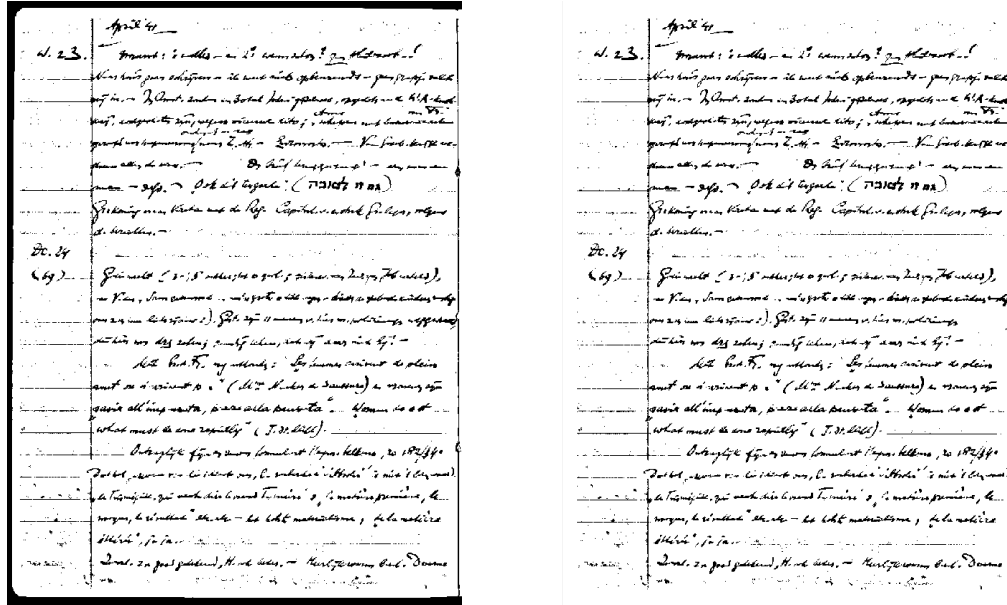
So far, we have obtained the position of each text line and the binary images that have been processed by using the morphological closing operation. To avoid the errors of computing connected components caused by touching characters from vertical directions, individual binary text lines are extracted as shown in Fig.3.5.

3.1.4 Computation of Connected Components

Fourthly, connected components (CCs) are computed within each individual binary text line that is extracted in the previous stage, which is an important processing step for textual components extraction. The operation of computing connected components can improve the structure of the background and simplify the following procedures by reducing noise in scanned images.

The process of noise reduction follows three steps: 1) computing connected components within an image; 2) finding the connected component having the maximal number of pixels that is considered as the black margin as shown in Fig.3.6 and the connected components having very small height or width that are considered as noise, at least not text information; 3) removing this noise from the image. For this purpose, we create the following criteria for removing proposed date blocks, which are determined through practical experiments:

1. $Num(pixels) = maximum$
2. $H(CC) < 20$ or $W(CC) < 20$
3. $Num(pixels) < 100$
4. $E(CC) < 0.1$



(A) Binary image before black margin removal. (B) Binary image after black margin removal.

FIGURE 3.6: Removing the black margin from a handwritten binary document image.

5. $D(CC) < 0.05$ or $D(CC) > 0.9$

$H(CC)$ and $W(CC)$ indicate the height and width of an extracted bounding box respectively. $E(CC)$ denotes the elongation, which is explained by equation 3.1. $D(CC)$ denotes the textual density described in equation 3.2, which is the ratio of the number of foreground pixels $F_n(CC)$ to the total number of pixels in the bounding box.

$$E(CC) = \frac{\min(H(CC), W(CC))}{\max(H(CC), W(CC))} \quad (3.1)$$

$$D(CC) = \frac{F_n(CC)}{H(CC) \cdot W(CC)} \quad (3.2)$$

3.1.5 Extraction of Proposed Date Blocks

Finally, the proposed date blocks within a handwritten document image are extracted based on the previous steps. Fig.3.7 shows an example image, where textual components are extracted and labelled by bounding boxes.

3.2 Positional Expectancy Model

After analysis of the date regions in our dataset, we found that dates are usually located on certain positions. For explanation convenience, we shall from now on refer to the

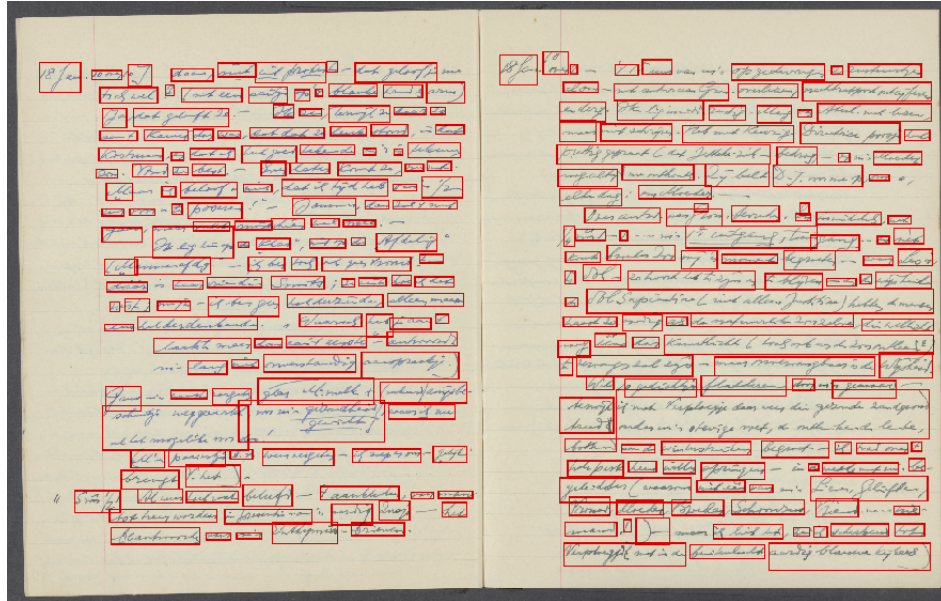


FIGURE 3.7: An example image for extracting proposed date blocks.

mid-point of a date region as the position of a date region. This phenomenon can be considered as additional knowledge for detecting date regions on handwritten document images, which is described as follows:

- An example image from our dataset is shown in Fig.3.8, where the date regions are manually labelled by rectangles and the contents are zoomed in beside the real date regions. We found that dates are more probable existing in the left regions of verso and recto pages than in other regions.
- All the mid-points of date regions in our dataset are shown in Fig.3.9. Obviously, dates emerge in some specific locations (e.g., left regions of each half page, top regions and top-left corner) more frequently than in other locations.

We conclude that dates may follow a particular position distribution over a page. Therefore, it is possible to propose a positional expectancy model to discriminate date regions from non-date regions based on their positions. The most important reason to use the proposed positional expectancy model is explained that: the positional expectancy model can compute the probability how much an unknown region is similar to a date region based on its position attribute, which can be viewed as the first discrimination between date and non-date regions.

A four-step approach has been used to design the proposed positional expectancy model. We first need to define the positional expectancy P_{dpe} of an unknown block: the positional expectancy is the probability of how much an extracted block is similar to a date region based on its position.

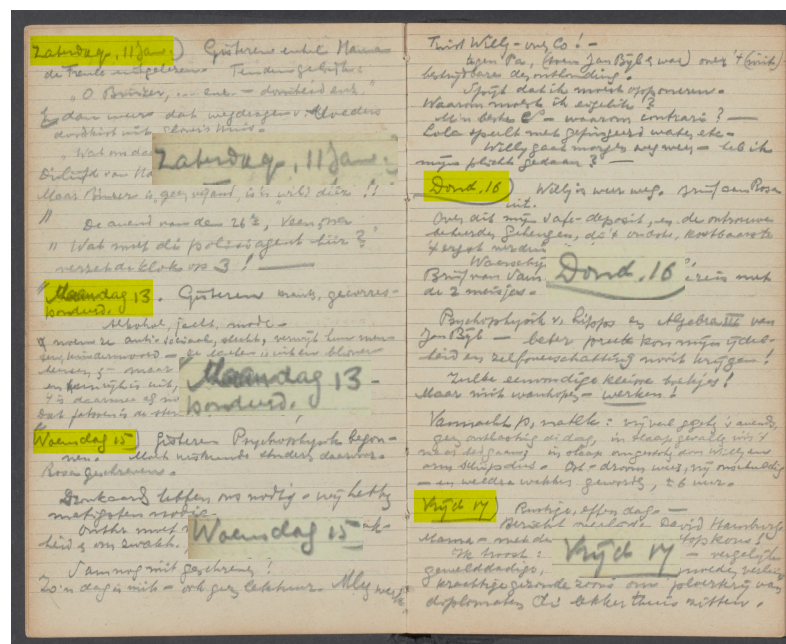


FIGURE 3.8: Example image for date regions and the corresponding contents in the DoLP dataset.

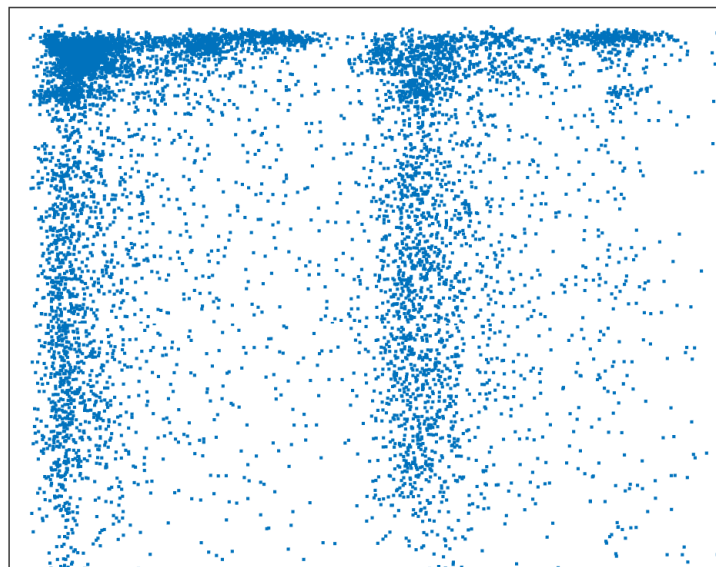


FIGURE 3.9: The density map of mid-points of date regions in the DoLP dataset. For full verso and recto page scans.

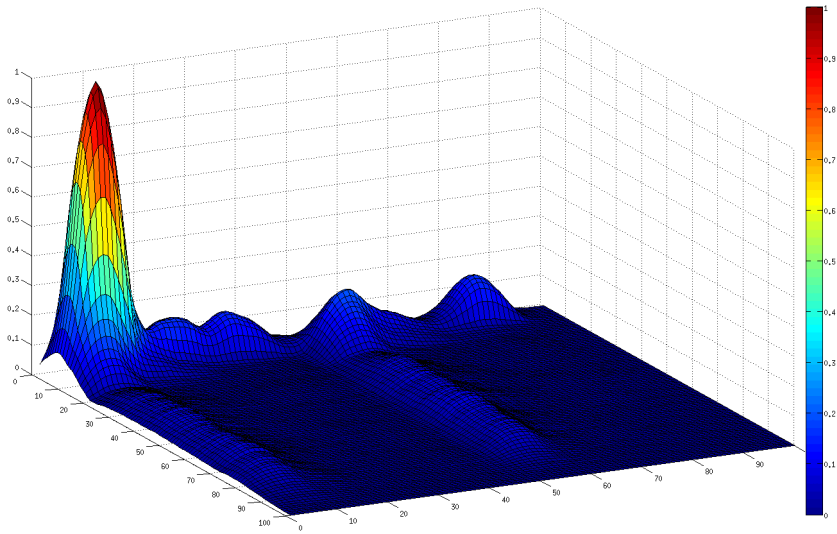


FIGURE 3.10: The density map of mid-point positions of date regions in the DoLP dataset shown in three dimensional space. For full verso and recto page scans.

Secondly, each page is normalized into $[0, 1]$ from left (0) to right (1), which is important for unifying the range of all scanned images.

Thirdly, we introduce two variables P_e and P_{ios} , which are used to compute the positional expectancy of an unknown block: 1) P_e denotes the empirical probability determining an extracted block is a date/not-date region, which can be computed through the position distribution of date regions in our dataset. Fig.3.10 shows the relation between date pattern's frequency of occurrence and their positions in three dimension. Obviously, many dates exist at the top-left corners and top horizontal regions of a page. 2) P_{ios} denotes how far an extracted block is from other extracted blocks. As shown in Fig.3.11, date regions are located on separated regions from other components both in horizontal and vertical directions. Therefore, we suppose that a date region can be viewed as an independent part from other components (e.g., titles, body paragraphs, comments, schematic drawings and author signatures) within document pages. The distances (Manhattan distance) are computed between an extracted block and its neighbours, which are used to measure independence of an extracted block.

Finally, the positional expectancy P_{dpe} can be computed by taking the multiplication of P_e and P_{ios} as shown in equation 3.3. To balance the effect of each part for the positional expectancy computation, P_e and P_{ios} are both normalized into $[0, 1]$.

$$P_{dpe} = N(P_e) * N(P_{ios}) \quad (3.3)$$

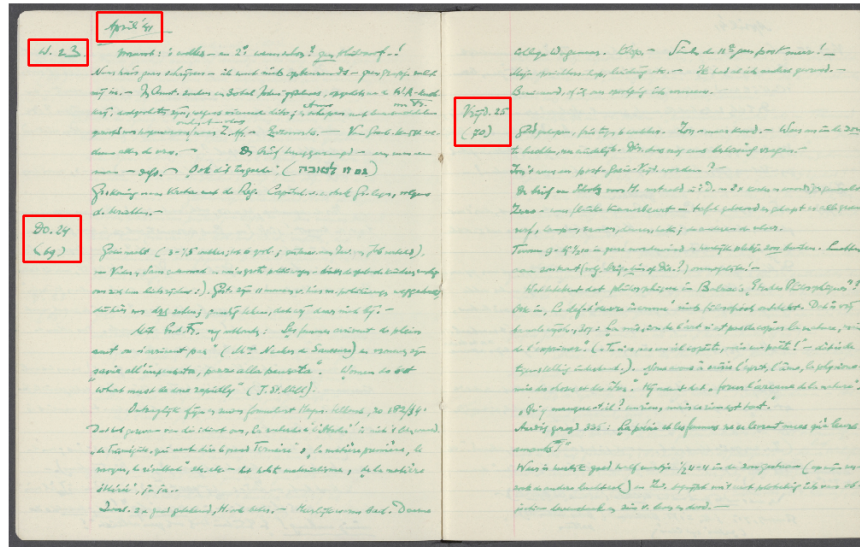


FIGURE 3.11: Date regions are isolated from their neighbours in horizontal or vertical directions.

3.3 Feature Representation and Classification

For feature representation and classification, we have attempted two different methods: 1) SVM-based classifier and 2) ConvNets model. First, we extract the HoG feature [14] from the proposed date blocks extracted from the pre-processing stage and then classify them into date or non-date class by using an SVM [14] classifier. Second, a ConvNet model is designed to detect date regions on handwritten document images, which contains both the processes of feature representation and classification in an end to end framework. The implementation details of both methods are described as follows.

3.3.1 SVM-Based Classifier

The SVM-based date classifier consists of two parts: a feature descriptor using the HoG feature and a non-linear SVM classifier by using the RBF kernel.

To obtain the HoG feature vectors that have identical dimensionality, the extracted blocks are first resized into $[112, 256]$, which is computed based on the size distribution of date regions in the DoLP dataset. Second, the HoG feature is extracted from each proposed date block using 9 orientations, which results in a 3472-dimensional feature vector. To reduce the complexity of computation, we select the cell size $N = 16$ in this thesis. Finally, the decision value $g(x)$ is computed by using equation 2.15, which is described in Section 2.3.1. Then we use the sigmoid function to compute the probability $P(x)$ of a sample is classified into date or non-date classes. We assume that $fApB =$

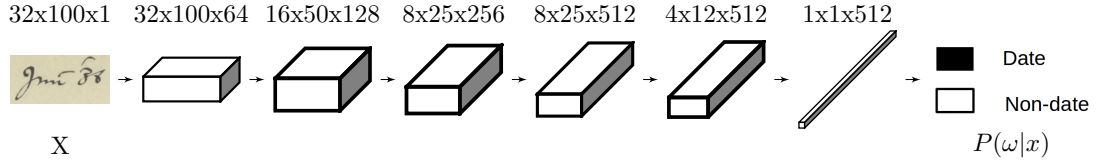


FIGURE 3.12: A schematic of the CNN used for date region detection from handwritten documents.

$g(x) * a + b$, where the parameters a and b are the slope and intercept parameters. Hence, $P(x)$ is computed by using equation 3.4. If the value of $P(x)$ is larger than 0.5, the unknown block belongs to the date class, otherwise it belongs to the non-date class.

$$P(x) = \frac{1}{1 + \exp(-fApB)} \quad (3.4)$$

3.3.2 ConvNets Model

The architecture of the ConvNets model in this thesis follows but is not the same as the text recognition ConvNets architecture described in [52]. The ConvNets model applied here consists of seven weight layers including five convolutional layers and two fully-connected layers. A schematic diagram of the used ConvNets model is shown in Fig.3.12. The filter size and number of filters selected in the convolutional layers are: $\{5,64\}$, $\{5,128\}$, $\{3,256\}$, $\{3,512\}$ and $\{3,512\}$ respectively. The first fully-connected layer has 512 units and the final fully-connected layer has two units (i.e., date and non-date). Each convolutional layer is followed by a Rectified Linear Unit (ReLU), which applies an element-wise activation function, $f(x) = \max(0, x)$, thresholding at zero. A 2×2 max pooling layer is used after all except for the third convolutional layer, which decreases the spatial size of the representation gradually so that it reduces the quantity of parameters and computations in the ConvNets model.

In addition, the fixed size of an input image to the ConvNet is a 32×100 grey-scale image, which is zero centred by subtracting the image mean. Zero-padding is applied to the input of each convolutional layer, which allows us to control the spatial size of the output volumes. Finally, the loss function used in the last fully-connected layer is a softmax function, which can be described by equation 3.5.

$$f(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (3.5)$$

3.4 Final Decision

As mentioned in Section 1.1, the research question in this thesis is how to classify a handwritten region as being a date region. Now, we combine features of a date pattern and its position together, which is described by equation 3.6:

$$Pd = P(d = 1|f, p) \tag{3.6}$$

where f denotes the features (e.g., shape, appearance or abstract features) of a date pattern and p denotes its position. We assume that f and p are independent from each other, hence equation 3.6 can be rewritten as:

$$Pd = P(d = 1|f) \times P(d = 1|p) \tag{3.7}$$

In equation 3.7, $P(d = 1|f)$ can be computed through equation 3.4 or the ConvNet model, and the second part $P(d = 1|p)$ is equal to P_{dpe} .

Finally, each extracted text block has a Pd value and then we sort this sequence from maximum to minimum. Finally, we select the top T blocks as date regions. The parameter T will be described in the next chapter.

Chapter 4

Experiments and Results

This chapter is divided into three parts. First, Section 4.1 provides an overview of the DoLP dataset used in this thesis, where we only focus on two types of data: 1) handwritten dates and 2) visually salient items on scanned document images. Second, three experiments are conducted to evaluate the performance of the 1) algorithm of block extraction, 2) positional expectancy model and 3) SVM-based classifier and ConvNets model for date regions detection respectively, which are described in Section 4.2. Finally, Section 4.3 presents the final evaluation of the proposed DateFinder method.

4.1 Dataset

To our best knowledge, there is no standard dataset that can be used to evaluate a detection method of handwritten date regions. The dataset in this thesis incorporates 53 volumes of diaries written in Dutch by the Dutch philosopher Leo Polak from 1905 to 1941, which is used for training and testing of the proposed DateFinder method. See Fig.4.1 for a sample scanned image in the DoLP dataset.

Since the proposed date detection method is trained by a supervised learning algorithm, all the date regions in the DoLP dataset are manually labelled as positive examples. The details of data labelling are described as follows.

4.1.1 Data Labelling

In this project, two types of data have been labelled: 1) handwritten dates and 2) visually salient items by using a Matlab program, whose graphic user interface (GUI) is shown in Fig.4.2. The user guidance is described as follows:

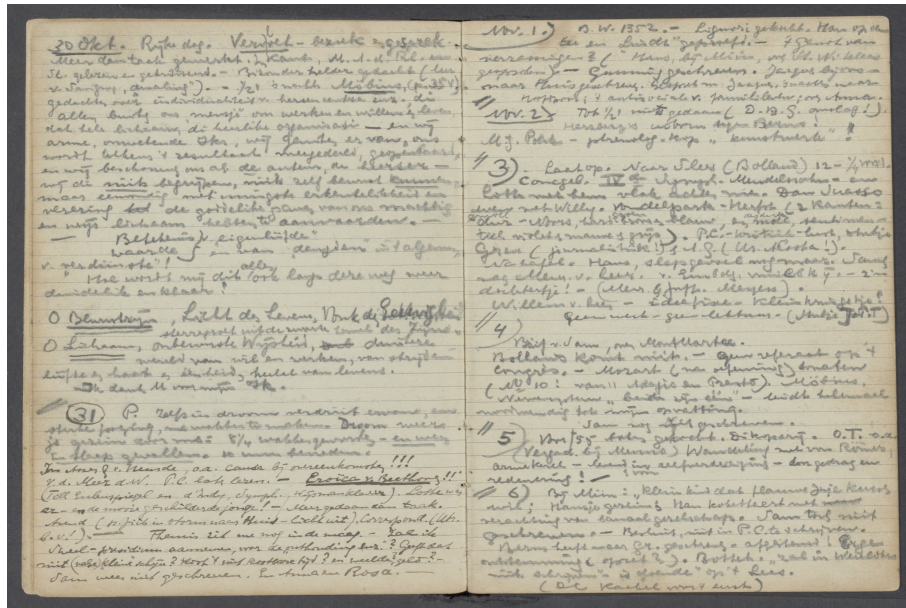


FIGURE 4.1: Example of a scanned image in the DoLP dataset.

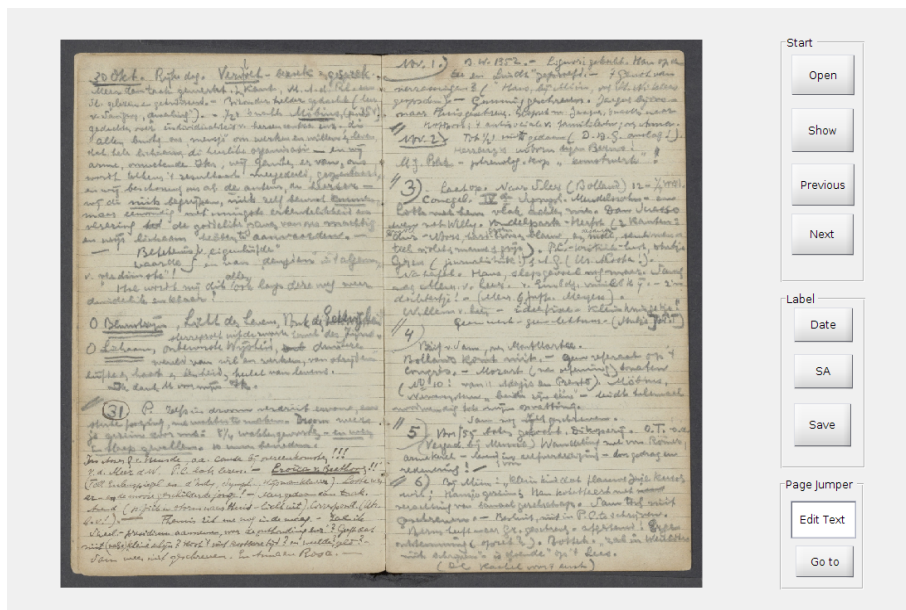


FIGURE 4.2: Matlab program for data labelling.

1. Select an image by using 'open' button.
2. Label data as date or visually salient item through 'Date' and 'SA' buttons.
3. 'Previous' and 'Next' buttons allow users to review the previous page or continue to label data in the next page. Also, users can apply 'page jumper' to any page they want.
4. When pressing the 'Save' button, data can be stored in a 'DF-data-xx.m' file, where 'xx' is the current date and time.

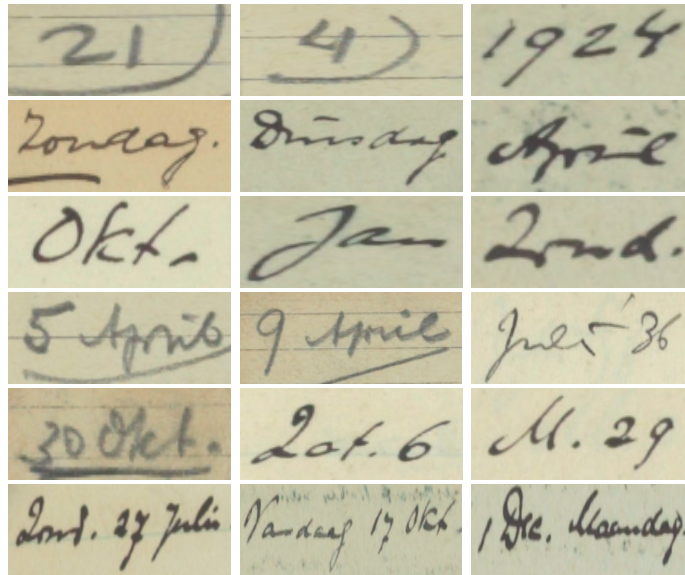


FIGURE 4.3: Six types of date formats in the DoLP dataset: 1) only numeral, 2) only word, 3) only abbreviation, 4) combination of numeral and word, 5) combination of numeral and abbreviation and 6) combination of numeral, word and abbreviation.

4.1.1.1 Date Information

As described in Chapter 1, dates can be in different formats in the DoLP dataset. Fig.4.3 shows some date patterns, where a date can contain only numerals, words, abbreviations or their combinations.

Since the objective of this project is to detect date regions on handwritten document images, the pages that only contain non-handwritten dates (i.e., created by machines) as shown in Fig.4.4 have been removed from the DoLP dataset.

4.1.1.2 Visually Salient Items

Apart from dates, there are many other visual items on the scanned images of the DoLP dataset. In this thesis, the visually salient items can be defined as the items that have significant properties, such as colour, shape, appearance or position on handwritten document images. Hence, the visual items collected in the DoLP dataset should be sufficiently different from their contexts so that they can attract the humans' attention directly.

For simple storage, we use letters to represent different types of visual items. For instance, the letter 'h' denotes that the labelled data is a drawing. See Table 4.1 for visual items and the codebook.



FIGURE 4.4: Scanned images only consist of non-handwritten dates, which should be removed from the DoLP dataset.

TABLE 4.1: Codebook and visual items in the DoLP dataset.

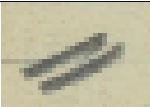

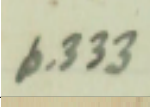
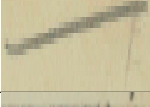
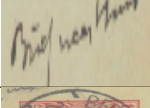

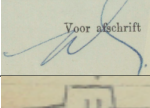
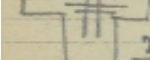
Code	Types	Patterns	Numbers
a	separator		2745
b	circled region		276
c	numeral		3478
d	underline		7895
e	comment		897
f	stamp		659
g	signature		35
h	drawing		21

TABLE 4.2: Attributes of dates and visual items for a few examples.

ID	D/S	x_1	y_1	W	H	Content
1	0	325	141	606	116	‘9 aug 1901 - 10 Juli 1902’
2	0	1572	139	679	134	‘9 aug 1901 - 10 Juli 1902’
3	0	1959	149	444	112	‘9 aug 1901’
3	0	2094	313	238	63	‘10 Juli 1902’
5	0	138	472	314	86	‘12 Sept 1901’
5	0	1403	587	297	83	‘8 Okt 01’
5	0	1363	1406	274	66	‘21 Okt 1901’
5	1	2179	1319	271	150	‘h’

4.1.1.3 Data Storage

Data are labelled by grabbing a rectangle, whose top-left corner point (x_1, y_1) and bottom-right corner point (x_2, y_2) are stored and used to indicate a data block. Table 4.2 shows how the data is stored. The first column denotes the index number of an image. In the D/S column, 0 indicates that the data in this row refers to dates and 1 indicates that the data refers to visually salient items except for dates. From the 3rd to 6th columns, the position of a data block (i.e., top-left corner point (x_1, y_1) , width and height) is stored. The last column denotes the content in a data block. For mathematical calculations, we use W to denote the width and H to denote the height of a block, which are described in equation 4.1 and 4.2.

$$H = y_2 - y_1 \quad (4.1)$$

$$W = x_2 - x_1 \quad (4.2)$$

4.1.2 Training, Validation and Test Datasets

There are 4825 scanned images in the DoLP dataset in total. Some of them are non-handwritten documents including drawings, newspapers and covers, which should be removed from our dataset. In each volume of the diary, 70 percent of the usable scanned images are selected randomly for training, 10 percent for validation and the rest of the scanned images are used in the test stage. Table 4.3 shows the number of these scanned images for different stages.

TABLE 4.3: Number of images of the DoLP dataset for training, validation and test.

Training	Validation	Test	Removed images
3088	772	965	554

4.2 Experiments and Results

In this section, three experiments are conducted to evaluate the performance of each part in the proposed DateFinder method.

1. Experiment 1: Input Image + Pre-processing
2. Experiment 2: Input Image + Pre-processing + SVM-based Classifier/ConvNets Model.
3. Experiment 3: Input Image + Pre-processing + Positional Expectancy Model + SVM-based Classifier/ConvNets Model.

In experiment 1, the performance of the algorithm of text block extraction is first evaluated by computing how many relevant date regions are extracted in the DoLP dataset. Experiment 2 compares the performance of the SVM-based classifier to the ConvNets model to determine which is the better method for detecting date regions on handwritten document images. Finally, we conduct experiment 3 and compare the results to experiment 2 to determine whether the proposed positional expectancy model is beneficial to improve the performance of the DateFinder method.

4.2.1 Experiment 1

Evaluating the performance of the proposed block extraction algorithm follows three steps. For simple explanation, we call real date blocks DBs (i.e., ground truth) and extracted blocks EBs (i.e., proposed date blocks). Also, we define N_i as the number of date blocks in scanned image i and M_i as the number of date blocks that find overlapping EBs in scanned image i . H is the set of scanned images, which contains date patterns. K is the number of elements in the set H .

First, text blocks are extracted by computing connected components and their positions are recorded. Second, we compute the overlap of each DB and all EBs within individual scanned images respectively. Initially, $M = 0$ and if a DB finds a corresponding block that has overlap equal or more than 50 percent with this DB in EBs, then $M_i = M_i + 1$.

TABLE 4.4: Evaluation of the proposed text block extraction algorithm.

Total dates	Extracted dates	Un-extracted dates	Recall
8794	8297	497	0.943

We use $R_i = M_i/N_i$ to denote the ratio of DBs which are included in EBs by using the proposed block extraction algorithm in scanned image i . Finally, we use P to denote the performance of the proposed block extraction algorithm, which can be described in equation 4.3:

$$P = \frac{\sum R_i}{K}, i \in H \quad (4.3)$$

The reason for this evaluation is twofold. Firstly, the objective of the proposed block extraction algorithm is to extract as many blocks containing dates as possible. Hence, recall of DBs is more important than precision in this case. Secondly, if a certain DB finds its corresponding EB, we stop the comparison between DB and EBs, which can reduce a lot of computation.

Table 4.4 shows the performance of the proposed date block extraction algorithm. 94.3% date regions in ground truth can be extracted by using our algorithm. The primary reason for this result (i.e., less than 100%) can be explained: some pixels of the characters consisting of dates are lost in the process of binarization, which leads to these characters changed into many small components, then the small components are removed through the process of noise reduction in the connected components computation stage. Therefore, it is difficult to ensure all the date regions in our dataset can be extracted as ‘date candidates’ for later classification.

4.2.2 Experiment 2

Experiment 2 compares the performance of the SVM-based classifier and ConvNets model for feature representation and date/non-date classification in the date regions detection task.

4.2.2.1 Test of SVM-Based Classifier

Due to the simpler structure and implementation of linear SVMs and better performance of non-linear SVMs, both modes of the SVM are tested in this section by using a 2-step method. First, 5-fold cross validation is applied in the training stage to determine

TABLE 4.5: Number of examples for the SVM-based classifier training and validation.

Types	Training	Validation
Positive	6943	903
Negative	11783	2072

TABLE 4.6: Evaluation of both Linear and Non-linear SVM classifier by using the validation dataset.

Types of SVM	Sensitivity	Specificity	Accuracy
Linear SVM	29.3% \pm 6.1%	92.4% \pm 7.1%	60.5% \pm 12.1%
Non-linear (RBF kernel) SVM	37.3% \pm 9.2%	95.6% \pm 8.7%	68.4% \pm 13.2%

the best parameters. We obtain the best $C_l = 2^{13}$ for the linear SVMs classifier and $C_{non} = 2^{12}$ and $\gamma_{non} = 2^{-11}$ for the non-linear SVM classifier.

Second, we compare the performance of the SVM classifier in different modes (i.e., linear and non-linear SVM classifier) based on three statistical measures: sensitivity, specificity and accuracy. The sensitivity is also called the true positive rate, which measures the proportion of positive examples that are correctly classified. The specificity is also called the true negative rate, which measures the proportion of negative examples that are correctly classified. Accuracy is the entire measurement both of positive and negative examples that are correctly classified.

The number of examples for the proposed SVM-based classifier for training and validation is shown in Table 4.5, where the examples in the validation dataset are used to compare the performance of linear and non-linear SVM classifiers. As described in Table 4.6, the non-linear (RBF kernel) SVM classifier is more powerful for detecting date regions on handwritten document images than the simple linear SVM classifier in all three measurements.

4.2.2.2 Test of ConvNets Model

Due to the binary (i.e., date or non-date) classification problem in this thesis, the last fully-connected layer outputs two scores, one of which indicates the probability that an input is classified as being a date region and the other one indicates that an input is classified as being a non-date region. A threshold $t = 0.5$ is set to classify an extracted block into date or non-date categories.

TABLE 4.7: Number of examples for the ConvNet model training and validation.

Types	Training	Validation
Positives	6943	903
Negatives	51863	5909

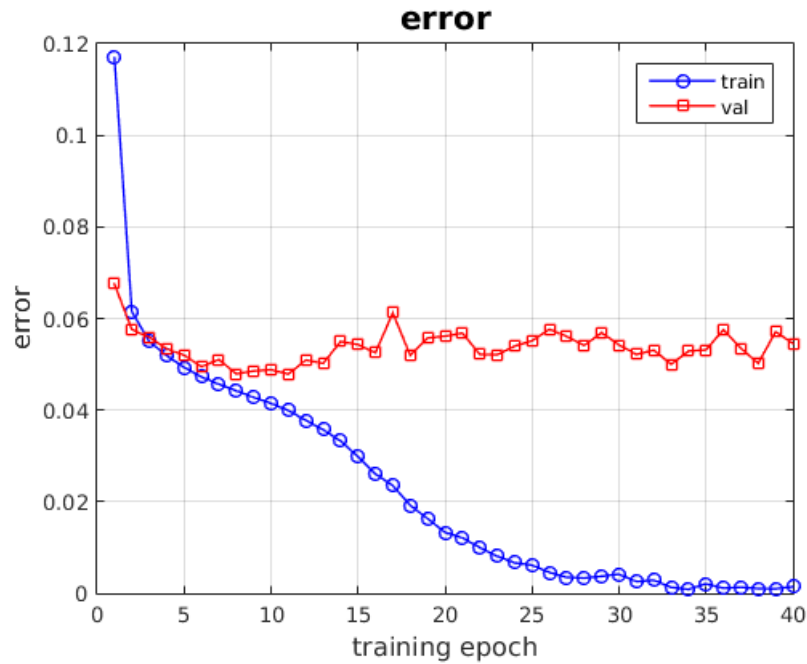


FIGURE 4.5: Classification error of ConvNets model both in training and validation stages from epoch 1 to 40.

TABLE 4.8: Evaluation of ConvNets model by using the validation dataset.

Type	Sensitivity	Specificity	Accuracy
ConvNets	46.7% \pm 9.2%	98.5% \pm 11.4%	94.3% \pm 14.7%

The number of examples for training the proposed ConvNet model is shown in Table 4.7, where we use the examples in the validation dataset to evaluate the performance of the ConvNet model. Fig.4.5 shows the classification error in both training and validation stages during 40 training epochs. The classification error of positive and negative examples decreases dramatically in the training stage. Finally, it reaches a low value around 0.006. The classification error fluctuates between 0.05 and 0.06 in the validation stage, which achieves 0.057 in epoch 40. Table 4.8 shows the results including sensitivity, specificity and accuracy of the proposed ConvNets model.

4.2.3 Experiment 3

In this thesis, the positional expectancy model is an important part in the proposed DateFinder method, which can measure how much a handwritten region is similar to a date region based on its position. In experiment 3, we add the positional expectancy model on the basis of experiment 2 and compare the results of both experiments, which aims to validate the feasibility of the proposed positional expectancy model. The results are described in the final evaluation of the proposed DateFinder method, which are shown in the next section.

4.3 Final Evaluation of DateFinder Method in Free Search

In this Section, we measure precision, recall and $F_{0.5}$ score for the final evaluation of the proposed DateFinder method in free search:

- Precision can be defined as the ratio of the number of correctly detected date blocks to the total number of extracted blocks.
- Recall can be defined as the number of correctly detected date blocks to the total number of date blocks in a scanned image.
- $F_{0.5}$ from best value at 1 and worst at 0 score is used to measure the accuracy in the test stage, which can be explained as a weighted average of the precision and recall.

To determine the parameter T used to control how many date regions are selected from the sequence of block candidates, we test $T = 5, 10, 15, 20$ respectively. The maximum ($T = 20$) is computed based on the distribution of the number of dates in each scanned image, since we found that all the scanned images contain less than 20 dates, which can guarantee high recall.

The performance of each method with different parameter T : 1) only using SVM-based classifier, 2) only using ConvNets model, 3) combination of positional expectancy model and SVM-based classifier and 4) combination of positional expectancy model and ConvNets model are shown in the Tables 4.9 - 4.12.

The final evaluation of each method is shown in Table 4.13. First, the ConvNets model shows better performance than the SVM-based classifier for detecting date regions on handwritten document images in the DoLP dataset, which can be concluded by comparing the first and second rows. Second, the average precision, recall and $F_{0.5}$ score

TABLE 4.9: Free search results by only using SVM-based classifier.

T	Recall	Precision	$F_{0.5}$ score
5	16.3% \pm 7.1%	13.6% \pm 6.9%	14.2% \pm 7.1%
10	20.2% \pm 7.0%	13.9% \pm 6.7%	14.4% \pm 6.9%
15	23.1% \pm 7.1%	13.6% \pm 6.6%	14.7% \pm 7.0%
20	25.9% \pm 7.0%	13.5% \pm 6.6%	14.9% \pm 7.0%

TABLE 4.10: Free search results by using positional expectancy model and SVM-based classifier.

T	Recall	Precision	$F_{0.5}$ score
5	40.6% \pm 10.8%	26.7% \pm 5.9%	28.5% \pm 6.2%
10	47.0% \pm 10.2%	21.5% \pm 3.4%	23.9% \pm 3.5%
15	51.1% \pm 10.5%	19.5% \pm 3.8%	22.1% \pm 3.8%
20	55.8% \pm 10.2%	18.3% \pm 4.2%	21.0% \pm 4.2%

TABLE 4.11: Free search results by only using ConvNets model.

T	Recall	Precision	$F_{0.5}$ score
5	17.1% \pm 4.4%	14.8% \pm 6.0%	15.1% \pm 5.8%
10	20.1% \pm 4.3%	14.5% \pm 6.1%	15.2% \pm 6.1%
15	22.0% \pm 9.1%	14.3% \pm 7.2%	15.2% \pm 7.4%
20	24.3% \pm 5.2%	14.3% \pm 7.1%	15.4% \pm 7.5%

TABLE 4.12: Free search results by using positional expectancy model and ConvNets model.

T	Recall	Precision	$F_{0.5}$ score
5	43.1% \pm 14.0%	31.4% \pm 9.7%	33.0% \pm 9.8%
10	46.1% \pm 11.2%	24.5% \pm 5.7%	26.7% \pm 5.7%
15	53.2% \pm 11.2%	22.6% \pm 8.8%	25.1% \pm 8.9%
20	53.5% \pm 11.4%	21.3% \pm 6.7%	24.0% \pm 7.0%

TABLE 4.13: Final evaluation of the free search performance of the proposed system using all methods.

Methods	Recall	Precision	$F_{0.5}$ score
SVM-based Classifier	$25.9\% \pm 7.0\%$	$13.5\% \pm 6.6\%$	$14.9\% \pm 7.0\%$
ConvNets Model	$24.3\% \pm 5.2\%$	$14.3\% \pm 7.1\%$	$15.4\% \pm 7.5\%$
Positional Expectancy Model and SVM-based Classifier	$40.6\% \pm 10.8\%$	$26.7\% \pm 5.9\%$	$28.5\% \pm 6.2\%$
Positional Expectancy Model and ConvNets Model	$43.1\% \pm 14.0\%$	$31.4\% \pm 9.7\%$	$33.0\% \pm 9.8\%$

are all improved by using the positional expectancy model, which can be demonstrated by comparing the first two rows and the last two rows. Finally, the combination of positional expectancy model and ConvNets model achieves the best performance in this experiment, which is shown in the last row of Table 4.13.

Chapter 5

Discussion

This chapter is divided into three parts. Firstly, Section 5.1 discusses the challenges of the DoLP dataset in this thesis. Secondly, we discuss the performance of the proposed ConvNets model and SVM-based classifier for detecting date regions on handwritten document images in Section 5.2.

5.1 Dataset Challenges

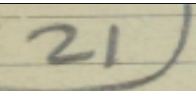
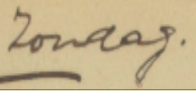
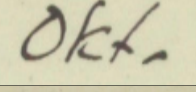
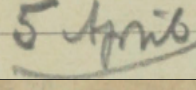
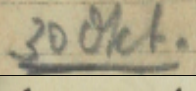
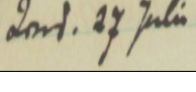
The materials in the DoLP dataset are challenging in five aspects: 1) different date formats, 2) undetermined position of dates, 3) low quality of images 4) touching characters and 5) mis-classification among date, numeral and letter. One or more of these challenges can affect the performance of the proposed DateFinder system, which are described as follows.

5.1.1 Different Date Formats

Dates can have different formats in the DoLP dataset. The main reason is that diaries can be viewed as informal documents, which allow writers to use any symbols ('I' and 'II'), characters ('a' and 'b'), numerals ('1' and '2') or words ('Monday' and 'Tuesday') to represent dates. When writing diaries in real life, writers might use a complete format including 'year', 'month' and 'day' at the beginning of a month and then use simple numerals to indicate the exact dates for convenience.

As shown in Table 5.1, date formats in the DoLP dataset are classified into six classes: 1) only numeral, 2) only word, 3) only abbreviation, 4) numeral and word, 5) numeral and abbreviation and 6) numeral, word and abbreviation.

TABLE 5.1: Different formats of dates in the DoLP dataset.

Types	Patterns
Only numeral	
Only word	
Only abbreviation	
numeral and word	
numeral and abbreviation	
numeral, word and abbreviation	

Some methods (e.g., sliding window) can search date patterns based on regular expressions of dates, which need a lot of computation. However, it is difficult to detect date regions by using a fixed expression of a date in the DoLP dataset.

5.1.2 Undetermined Position of Date Regions

There are few constraints for date positions, which can be considered as an independent part in a diary. Dates are usually located on the corners of a page in the DoLP dataset, but these can also be at the front of a paragraph. See Fig.5.1 for an example page showing different date positions.

Therefore, we propose the positional expectancy model by computing the distribution of date regions in our dataset, which aims to measure how much an extracted block is similar to a date region based on its position.

5.1.3 Low Quality of Images

The low quality of scanned images can cause loss of information or produce noise in some stages (e.g., binarization). Although some operations are performed to reduce the noise and reconstruct the shape of characters in the pre-processing stage, it can still affect the accuracy of the computation of connected components, which consequently affects the performance of extracting proposed date blocks.

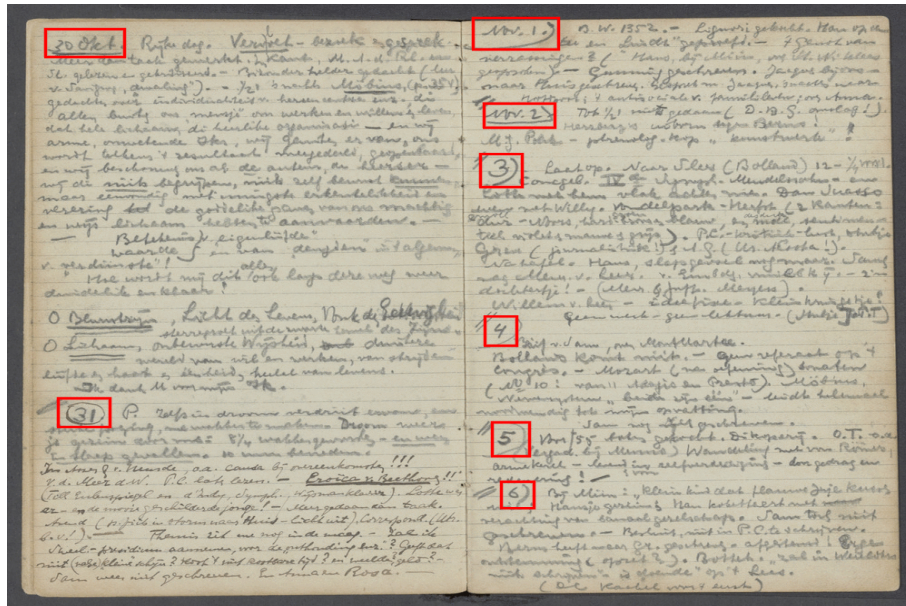


FIGURE 5.1: Undetermined positions of dates in the DoLP dataset.

5.1.4 Touching Characters

Characters usually touch their neighbours (i.e., stroke connection) in horizontal directions (i.e., left and right sides) in the DoLP dataset, which makes it difficult to accurately segment connected components for later classification. Also, touching characters can exist in vertical directions due to unintentional writing, which can affect the accuracy of extracting proposed date blocks. Fortunately, we have applied a text line segmentation algorithm called seam carving [48] for touching characters in vertical directions, which allows us to compute connected components in individual text lines.

5.1.5 Mis-classification among Date, Numeral and Letter

Mis-classifications among date, numeral and letter is another difficult challenge for detecting date regions on handwritten document images. For instance in Fig.5.2, the symbol '9' can be used to denote a date or a numeral in diaries and its appearance is very similar to the character 'g' in handwritten documents, which can result in mis-classification of date/non-date patterns.

Also, another mis-classification occurs between abbreviations of a date and parts of a word. For example, the abbreviation 'Dec' representing 'December' is the same as the part 'Dec' of the word 'Decimal'. It is possible to extract a block containing 'Dec' from the word 'Decimal' in the computation of connected components, which leads to mis-classification of date/non-date classifier.

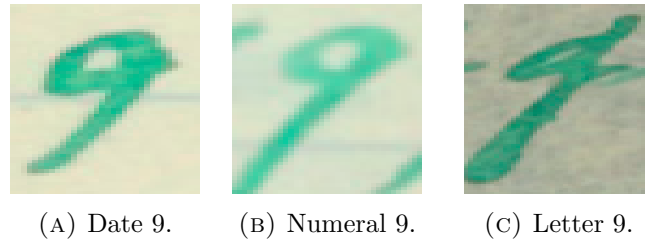


FIGURE 5.2: Similar patterns from different classes (i.e., date, numeral and letter).

To solve the problem of unreliability of the classifier that is trained only based on date and non-date patterns as described above, additional knowledge (e.g., contextual knowledge) is needed to help the detection system make an accurate classification decision.

The aforementioned challenges increase the difficulty of detecting date regions on handwritten document images in the DoLP dataset. To overcome this difficulty, more robust algorithms should be used for feature representation and date/non-date classification. In addition, additional knowledge (e.g., positions of dates) can be used to improve the performance of the proposed DateFinder method.

5.2 ConvNets Model and SVM-based Classifier

The proposed ConvNets model is more robust than the SVM-based classifier in terms of the performance of detecting date regions in this thesis. The main reason is that the ConvNets model has the ability to extract high-level features from input patterns through using many layers (convolutional layers, pooling layers and fully connected layers), but the HoG feature extraction method can only extract the features such as text shape and appearance. In addition, as another advantage of the ConvNets, the features in each layer can be automatically detected and learned from input. It is not necessary to design these features by hand [26]

In contrast to the ConvNets model, the SVM-based classifier has the advantages of including more time-saving training, less computation and a simpler structure than the ConvNets model. Also, there are less parameters in the SVM-based classifier than the ConvNets model. To obtain the optimal ConvNets model, many system parameters are needed to be modified, which will take a lot of time.

Chapter 6

Conclusion and Future Work

In this chapter, we first give a conclusion for this thesis in Section 6.1. Second, Section 6.2 describes the future work, which includes the optimization and extension of the proposed DateFinder method and the additional knowledge models for detecting date regions.

6.1 Conclusion

For humans, it is effortless to swiftly scan a text page for a target by using directed eye movements. However, when a computer system deals with target detection in big data (i.e., a large number of text images), the performance can be affected by searching time and complexity of computation. One popular solution is to detect field-based information rather than processing the entire image, which can decrease computation dramatically.

As described in this thesis, there are many visual items that have specific properties including appearance, shape, spatial position and color in handwritten documents, which make them stand out from their contexts. Thus, it is possible to extract the fields of these visual items based on their specific properties. Among these visual items in handwritten documents, dates play an important role in many documents (e.g., bank cheques, letters, postal mails, bills and diaries), which can provide clues of time-related information for readers.

To answer the research question in this thesis: *“How to detect date regions on handwritten document images?”*, we propose a method called DateFinder. Firstly, the operations of pre-processing are performed on the original scanned images, which aim to extract appropriate proposed date blocks. Secondly, a positional expectancy

model is used to measure how much an extracted block is similar to a date based on its position. Thirdly, we apply the ConvNets model to compute the probability of an extracted block is a date region, which has shown better performance than the SVM-based classifier in practical experiments. Finally, we combine the scores of the positional expectancy model and ConvNets model to make the final decision.

The proposed positional expectancy model has the ability to measure how much an extracted region is similar to a date region in the DoLP dataset based on its position. The basis behind this model is that dates exist in some specific locations (e.g., left regions of each half page, top regions and top-left corner) more frequently than in other locations. As the additional knowledge for date regions detection, the proposed positional expectancy model indeed improves the performance of the DateFinder method by assigning date blocks higher scores and non-date blocks lower scores.

This project can be viewed as a basis line for detecting date regions on handwritten document images. Although we have obtained encouraging results for detecting date regions in the DoLP dataset, there are still many things that can be researched, which are described in the next section.

6.2 Future Work

6.2.1 DateFinder Optimization

For the optimization of the proposed DateFinder method, many parameters should be first modified both in the positional expectancy model and the ConvNets model. Second, it is still possible to improve the performance of the ConvNets model through using more training examples (i.e., both positive and negative examples), increasing the training time, increasing the number of layers [53] and using dropout [40] for avoiding overfitting.

6.2.2 Additional Model for Detecting Date Regions

As described in Table 4.13, the proposed positional expectancy model has shown its worth for detecting date regions in the DoLP dataset. In order to improve the performance of the proposed DateFinder method, we can design other additional models based on the properties of dates. For instance, most of dates are adjacent to the separators, such as single or double slash, semicircle or circle, underline or their combinations in the DoLP dataset. Fig.6.1 shows some examples of the dates and their separators.

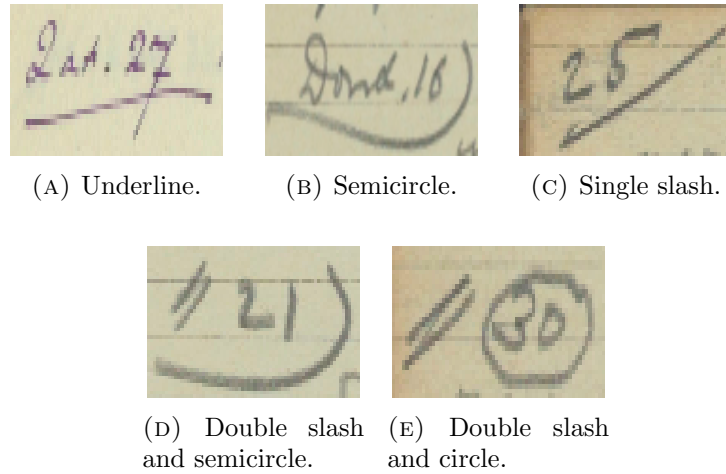


FIGURE 6.1: Examples of the dates and their separators in the DoLP dataset.

Therefore, it is possible to propose an additional model to measure that whether a proposed date block has an adjacent separator. If a proposed date block has an adjacent separator, it is more probable a date.

6.2.3 DateFinder Extension

The extension of the proposed DateFinder method can be explained in twofold. First, it is possible to extend the DateFinder method for recognizing dates on handwritten document images, which can be viewed as an important future work in this thesis. For instance, we can propose a handwritten dates recognition system by using a three-step method: 1) detecting the date regions by using the proposed DateFinder method, 2) segmenting the date patterns into characters and 3) using a classifier to classify the characters into 10 numeral categories (i.e., 0 to 9), 12 month categories (i.e., Januari to December) and 7 week categories (i.e., Maandag to Zondag). Second, there are many visual items existing the scanned images in the DoLP dataset, such as separators, author markings, schematic drawings, special symbols and glyphs, which can make readers understand the underlying meaning and origin of handwritten documents. The items mentioned above usually have specific properties, which make humans distinguish them from other visual items directly. Thus, it is possible to detect these kinds of visual items on scanned images in the DoLP dataset based on their specific properties.

In the DateFinder method, the positional expectancy model is generated based on the DoLP dataset, which is a barrier for generalization of the DateFinder method to other handwritten document datasets. How to make the proposed DateFinder usable in other datasets is a future consideration.

Bibliography

- [1] Thomas M Breuel. The ocrpus open source OCR system. In *Electronic Imaging 2008*, pages 68150F–68150F. International Society for Optics and Photonics, 2008.
- [2] Tijn Van der Zant, Lambert Schomaker, and Koen Haak. Handwritten-word spotting using biologically inspired features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1945–1957, 2008.
- [3] Lambert Schomaker. Design considerations for a large-scale image-based text search engine in historical manuscript collections. *Information Technology*, 59, 4 2016. ISSN 2196-7032.
- [4] Jean-Paul van Oosten and Lambert Schomaker. Separability versus prototypicality in handwritten word-image retrieval. *Pattern Recognition*, 47(3):1031–1038, 2014.
- [5] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [6] Simon Thomas, Clement Chatelain, Laurent Heutte, and Thierry Paquet. Alphanumeric sequences extraction in handwritten documents. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pages 232–237. IEEE, 2010.
- [7] Guillaume Koch, Laurent Heutte, and Thierry Paquet. Numerical sequence extraction in handwritten incoming mail documents. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 369–373. IEEE, 2003.
- [8] Clément Chatelain, Laurent Heutte, and Thierry Paquet. Segmentation-driven recognition applied to numerical field extraction from handwritten incoming mail documents. In *Document Analysis Systems VII*, pages 564–575. Springer, 2006.
- [9] Ching Y Suen, Qizhi Xu, and Louisa Lam. Automatic recognition of handwritten data on cheques—fact or fiction? *Pattern Recognition Letters*, 20(11):1287–1295, 1999.

-
- [10] Ratna Mandal, Partha Pratim Roy, and Umapada Pal. Date field extraction in handwritten documents. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 533–536. IEEE, 2012.
- [11] Ranju Mandal, Partha Pratim Roy, Umapada Palz, and Michael Blumenstein. Date field extraction from handwritten documents using HMMs. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 866–870. IEEE, 2015.
- [12] Qizhi Xu, Louisa Lam, and Ching Y Suen. A knowledge-based segmentation system for handwritten dates on bank cheques. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 384–388. IEEE, 2001.
- [13] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.
- [14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, 2005.
- [15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [17] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188–194, 2005.
- [18] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.
- [19] Ali Borji, Dicky N Sihite, and Laurent Itti. Salient object detection: A benchmark. In *European Conference on Computer Vision 2012*, pages 414–429. Springer, 2012.
- [20] Ali Borji, Dicky N Sihite, and Laurent Itti. Quantitative analysis of human-model agreement in visual saliency modeling: a comparative study. *Image Processing, IEEE Transactions on*, 22(1):55–69, 2013.
- [21] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, 2013.

- [22] Maurizio Corbetta and Gordon L Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3):201–215, 2002.
- [23] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.
- [24] A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.
- [25] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel Methods in Computational Biology*, pages 35–70, 2004.
- [26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [27] Hyeon-Joong Yoo. Deep convolution neural networks in computer vision. *IEEE Transactions on Smart Processing & Computing*, 4(1):35–43, 2015.
- [28] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013.
- [29] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.
- [30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.
- [31] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 196–201. IEEE, 2011.
- [32] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.
- [33] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *Acoustics, Speech*

- and *Signal Processing (ICASSP)*, 2013 *IEEE International Conference on*, pages 8614–8618, 2013.
- [34] Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168–174, 2013.
- [35] T Ciodaro, D Deva, JM De Seixas, and D Damazio. Online particle detection with neural networks based on topological calorimetry information. In *Journal of Physics: Conference Series*, volume 368, page 012030. IOP Publishing, 2012.
- [36] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.
- [37] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.
- [38] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.
- [39] Ilya Sutskever, Oriol Vinyals, and Le Quoc. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [40] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [41] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.
- [42] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009.
- [43] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106, 1962.
- [44] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [45] Yann LeCun, LD Jackel, L Bottou, A Brunot, C Cortes, JS Denker, H Drucker, I Guyon, UA Muller, E Sackinger, et al. Comparison of learning algorithms for

- handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60, 1995.
- [46] Margarita Osadchy, Yann Le Cun, and Matthew L Miller. Synergistic face detection and pose estimation with energy-based models. In *Toward Category-Level Object Recognition*, pages 196–206. Springer, 2006.
- [47] Margarita Osadchy, Yann Le Cun, and Matthew L Miller. Synergistic face detection and pose estimation with energy-based models. *The Journal of Machine Learning Research*, 8:1197–1215, 2007.
- [48] Nikolaos Arvanitopoulos and Sabine Susstrunk. Seam carving for text line extraction on color and grayscale historical manuscripts. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 726–731. IEEE, 2014.
- [49] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [50] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.
- [51] Jean Serra. *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.
- [52] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, pages 1–20, 2014.
- [53] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.