Unsupervised Feature Learning for Writer Identification

A Master's Thesis
Submitted to the Faculty of the Escola Tècnica
d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Student:
Eduard Pallàs Arranz

Advisors:
Stefan Fiel
Manuel Keglevic
Robert Sablatnig
and
Josep R. Casas

In partial fulfillment
of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Barcelona, July 2018

Title of the thesis: Unsupervised feature learning for writer identification

Author: Eduard Pallàs Arranz

Abstract

Our work presents a research on unsupervised feature learning methods for writer identification and retrieval. We want to study the impact of deep learning alternatives in this field by proposing methodologies which explore different uses of autoencoder networks.

Taking a patch extraction algorithm as a starting point, we aim to obtain characteristics from patches of handwritten documents in an unsupervised way, meaning no label information is used for the task. To prove if the extraction of features is valid for writer identification, the approaches we propose are evaluated and compared with state-of-the-art methods on the ICDAR2013 and ICDAR2017 datasets for writer identification.

Dedication:

This thesis is dedicated to the loved ones waiting at home, to Giulia, and to all those who have made Vienna an unforgettable experience.

Acknowledgements

First of all, I would like to sincerely thank Josep Ramon Casas for all the support and dedication of all these years, and also during the development of the thesis.

I would also like to sincerely thank Stefan Fiel and Manuel Keglevic for the guidance and supervision throughout this study, as well as Prof. Robert Sablatnig and the Computer Vision Lab for giving me the opportunity to work with them during my stay in Vienna.

Finally, to all my family and friends, thank you for your understanding and encouragement, which have been essential for the completion of the thesis.

Revision History

Stefan Fiel (SF), Manuel Keglevic (MK), Robert Sablatnig (RS), Josep Ramon Casas (JRC), Eduard Pallàs (EP)

Revision	Date	Author(s)	Description
1.0	23.04.18	EP	Creation
2.0	28.06.18	JR	First revision
3.0	10.07.18	SF,MK	Revision
3.1	13.07.18	MK	Revision
3.2	15.07.18	JR	Revision
4.0	16.07.18	JR	Final revision

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The first chapter of the thesis presents an overall view of the contents of the document. Sections within the chapter include motivation, context, objectives, organization plan and document structure.

"You don't write because you want to say something,
you write because you have something to say."

- F. Scott Fitzgerald

1.1 Motivation

Writer identification and writer retrieval are behavioral handwriting-based solutions which address recognition of the authorship of handwritten documents. The first concept returns the identity of the documents' writer, in which a graphical query is used to find the closest match against a labeled database. In contrast, writer retrieval aims to return a ranking of the documents in the dataset according to the similarity of the handwriting. Figure 1.1 shows the difference between the two concepts.

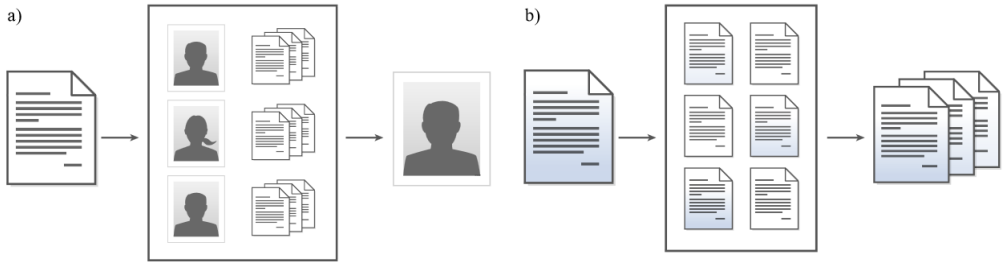


Figure 1.1: Illustration of the difference between writer identification (left) and retrieval (right). Image taken from [15].

The tasks of identification and retrieval encounter difficulties, some of which are depicted in Figure 1.2. Challenges to face include variations in handwriting styles, languages, character sets, layout or legibility among others. Moreover, the documents from the same author also present high variability in handwriting, such as writing characters in different sizes, changing writing utensils or document layout, which makes retrieval harder to achieve.

Therefore, we intend to identify specific characteristics from writers which can be robust against the aforementioned changes. Unsupervised deep learning approaches have been proven to be able to learn useful representations as for instance in P. Vincent et al. in [43]. Furthermore, Y. Netzer et al. [33] provided robust features for images in demanding situations using the same algorithms. Considering these successful approaches, we present an unsupervised learning approach based on the use of autoencoders, to study new feature extraction methods for writer identification and retrieval.

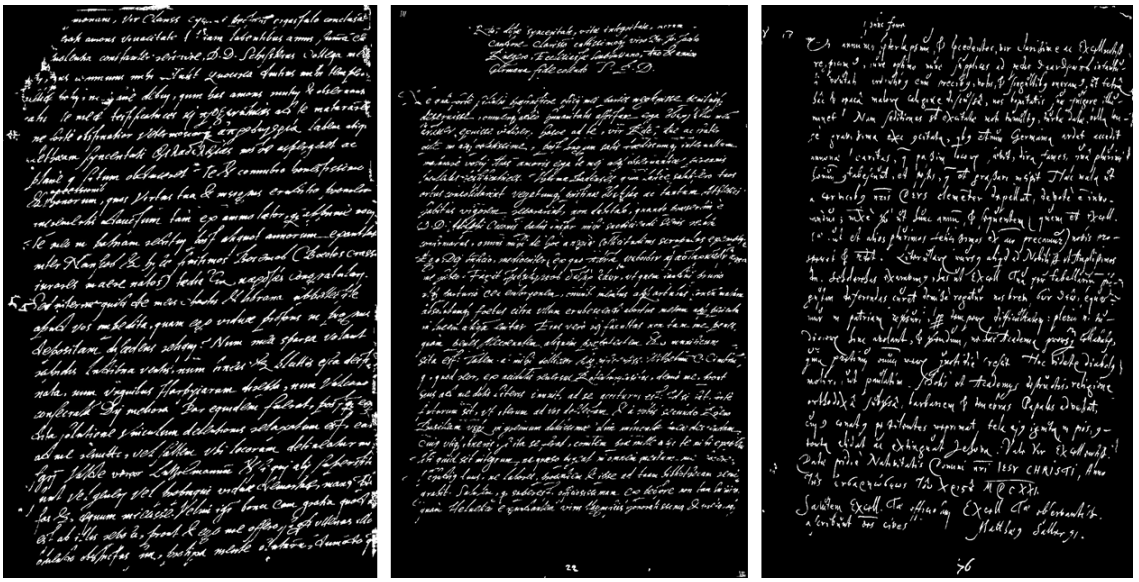


Figure 1.2: Binarized images from the ICDAR 2017 dataset [14]. First and second images from left to right, which present different layouts and character sizes, correspond to the same writer. The image from the right corresponds to a second writer.

1.2 Context

The Computer Vision Lab (CVL) has hosted the development of the thesis in an Erasmus+ mobility program. The research group is part of the Institute of Visual Computing & Human-Centered Technology, Faculty of Informatics, at the TU Wien, Austria.

Document Analysis is one of the theoretical backbones at CVL, whose contributions have gone as far as actively participating on Document Analysis competitions, taking part of the European Union Horizon 2020 READ project, among others. Researchers from CVL have a large background on the subject, from which research on writer identification plays an important role and is still a topic being studied. Within this context, the thesis continues the work of [8], in which CVL researchers contributed, with new unsupervised learning solutions.

1.3 Objectives

Given the motivation and context in which we find ourselves, we state that the objectives we want to achieve during the course of the project are:

Learn how to use a deep learning framework. We intend to master the basics of a deep learning framework to later acquire some more advanced skills.

Study and understand state of the art methods of writer identification and each of their contributions which provides a helpful overview of the field up to nowadays, and lets us see which ways have not been yet explored and what approaches to avoid.

Develop new ways to extract useful features from handwritten image-based samples for identification and retrieval, focusing on deep learning approaches and evaluate them objectively. We aim to compare against state of the art methods using well-known reference datasets which we will detail in the next chapter.

1.4 Document structure overview

In this section we depict the main structure of the thesis. Starting with motivation and objectives to know where the project is leading, the next chapters of the dissertation are organized as follows:

Chapter 2 gives an overview of the state of the art in writer identification. We showcase articles, competition approaches and recent works on the topic as well as popular benchmarking datasets.

Chapter 3 explains the procedure we followed to fulfill the objectives. We describe the main concepts of the approaches we explore, and define the methods we developed during the thesis.

Chapter 4 defines the metrics used to evaluate our methods, describes the experiments and displays the results obtained from those, later performing an analysis.

Chapter 5 contains the end conclusions and future work proposals for later studies.

Finally, an estimation of the costs of the development of the project can be found in Chapter 6.

1.5 Organization plan

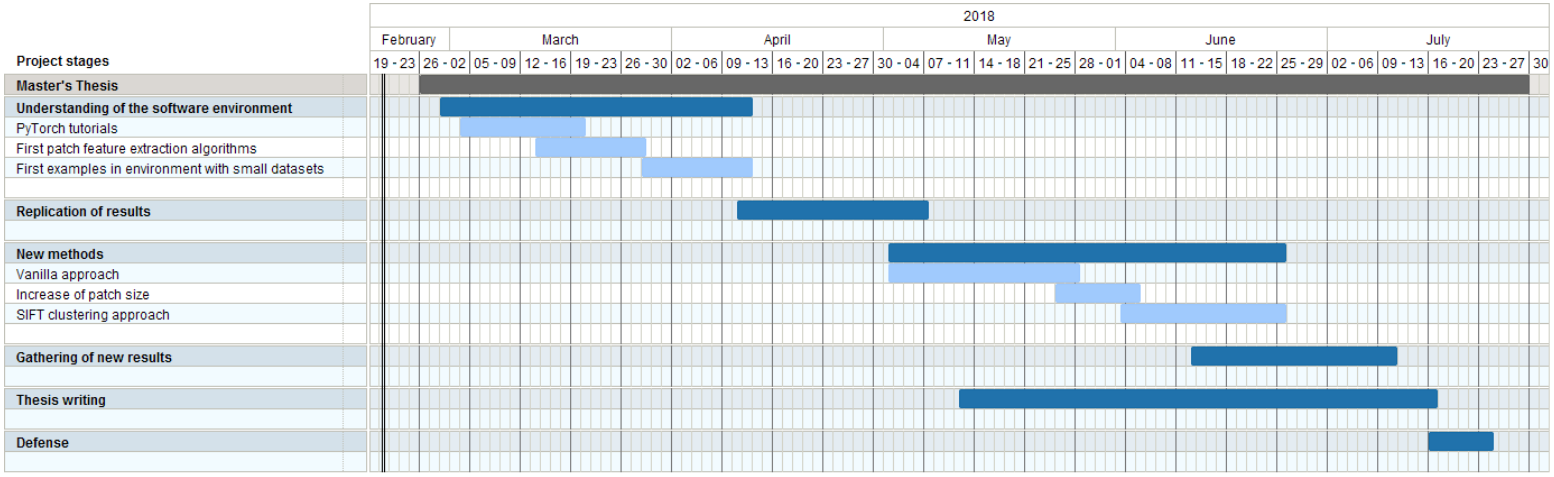


Figure 1.3: Gantt diagram of the organization plan of the thesis

Chapter 2

State of the art

This chapter contains a description of the state of the art in writer identific ation. We showcase benchmarking datasets and last studies on the field.

"Let us study things that are no more.
It is necessary to understand them,
if only to avoid them."

- Victor Hugo, Les Misérables

2.1 Document analysis

Earliest works to be considered as the beginning of document analysis span from studies on OCR (Optical Character Recognition), which is the conversion of images of typed, handwritten or printed text into machine-encoded text. We can find initial contributions before the 1990s like [30]. Furthermore, some surveys [41] from the same dates show a growing interest of academics in this field, which also takes special interest in handwriting [35]. Creation of formal workshops, conferences and competitions promoted an increase of studies related to the topic, and provided benchmarking tools such as state of the art datasets (which we explore in 2.2) and metrics (see 4.1) to have objective and quantitative manners to evaluate improvement. Some examples of this conferences still active are the International Conference on Document Analysis and Recognition (ICDAR), International Conference on Frontiers in Handwriting Recognition (ICFHR) and Document Analysis Systems (DAS).

Partly thanks to the research dissemination efforts in these conferences, several fields of interest developed, each focusing on obtaining different types of information from documents. Some examples of research competitions are: text recognition [13][40], keyword spotting [38], image binarization [37][36], document layout recognition [11], text reading (image to text) [46], and writer identification [26][14], which is the field in which we focus.

2.2 Benchmarking datasets

Common metrics and datasets allow objective comparison of methods. In this section we review three datasets we used to evaluate our approaches. Evaluation metrics are explained in section 4.1 of chapter 4.

2.2.1 MNIST dataset

The MNIST dataset [24] is a widely known database of handwritten digits, being a subset of a larger set available from NIST. It is composed of a training set of 60,000 examples, and a test set of 10,000 examples. The dataset is composed of 28x28 images containing handwritten digits from 0 to 9, which have been size-normalized and centered in a fixed-size image like seen in Figure 2.1.

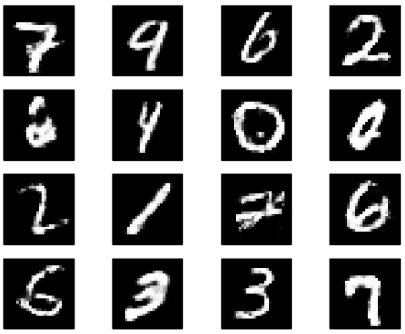


Figure 2.1: Sample from MNIST dataset

The database is a common starting point for beginners to try machine learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. The use of MNIST has been required to develop basic PyTorch skills and to understand the new software environment. No results or further investigation have been done once the proper skills had been acquired. Furthermore, images of the digits resemble the patches we process for writer identification in following experiments, which makes it suitable for trying new algorithms.

2.2.2 ICDAR 2013 dataset for writer identification

The 12th edition of the International Conference on Document Analysis and Recognition (ICDAR) hosted the "ICDAR2013 Competition on Writer Identification" [26] providing a specific dataset for such competition. Composed of images of handwritten samples, the database contains texts in English and Greek from 250 authors. Each one of them contributes with 4 fragments of text (2 in English and 2 in Greek), which make a total of 1000 images. Figure 2.2 showcases 4 pages from the dataset.

Despite the existence of larger and more complex databases we can also find artifacts which can make algorithm comparison unclear. Dirty or darker paper, spots, erased ink and many other problems avoid a fair evaluation of new methods, in contrast with the simplicity and cleanness of ICDAR2013.

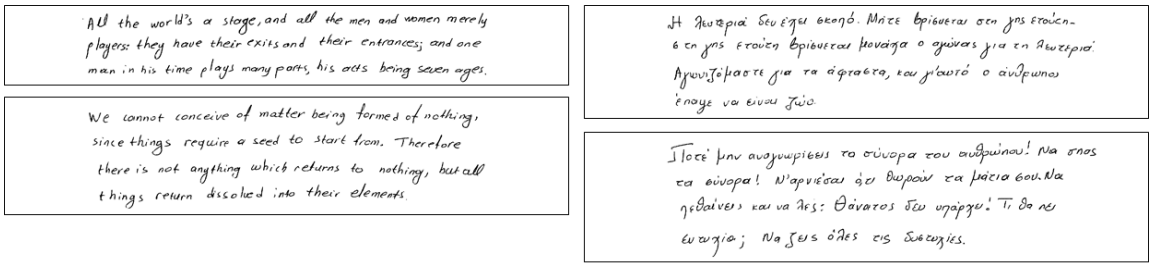


Figure 2.2: Four pages of the same writer from ICDAR2013 dataset. Left 2 in English, right 2 in Greek.

2.2.3 ICDAR 2017 dataset for writer identification

Similarly to the previous competition, the "ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI)" presented another dataset for authorship recognition for handwritten documents.

The dataset is composed by 4782 handwritten pages from 1114 different writers. Documents originating from 13th to 20th century from the digital archive of the Universitatsbibliothek Basel are divided into test and train. The test set consists of five document images per individual writer and three document images are available for training, resulting in 3600 and 1182 pages for test and train respectively. Note that no writer of the training set has any page in the test set.

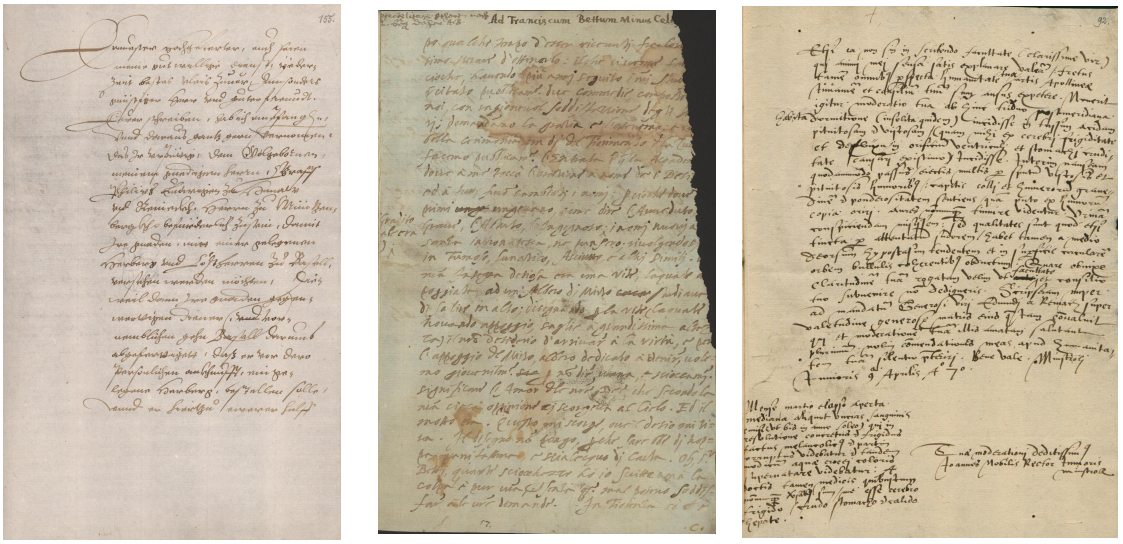


Figure 2.3: Three sample pages of the Historical-WI dataset

Dataset	Number of writers	Pages per writer	Total pages
ICDAR 2013 [26]	250 50 train/200 test	4	1000 200/800
ICDAR 2017 [14]	1114 394 train/750 test	3 train/5 test	4782 1182/3600

Table 2.1: Datasets comparison

We consider this to be a more challenging dataset than ICDAR 2013. The current dataset consists of historical documents which do not have a uniform background, the text lines often overlap and words differ among pages. In contrast, ICDAR 2013 was generated in a restricted environment providing characteristics such as uniform background and non-overlapping text lines. Table 2.1 delivers a comparison of the ICDAR datasets we will later use as benchmarks.

2.2.4 Other datasets

There are more datasets which we have not used that offer interesting characteristics for future purposes. Some of those are:

- ICDAR 2011 Writer Identification Contest dataset [27] consists of 208 documents from 26 writers, previous to ICDAR 2013.
- International Conference on Frontiers in Handwriting Recognition ICFHR 2012 [28], which created a dataset with the help of 100 writers that were asked to copy four parts of text in two languages (English and Greek), like seen in ICDAR 2013.
- DIVA-HisDB Historical Document Image Database [39], published in ICFHR 2016, the dataset is a precisely annotated large dataset of challenging medieval manuscripts. It consists of three medieval manuscripts, 50 pages each, resulting of in total 150 pages.
- CVL-DataBase [21] includes 7 different handwritten texts (1 German and 6 English texts) and 311 different writers.
- HACDB Handwritten Arabic Characters Database [22], used for automatic character recognition, contains 6.600 shapes of handwritten characters written by 50 people.

2.3 State of the art in writer identification

A wide variety of writer identification techniques have been reported in the literature. In this section we present an overview in state-of-the-art methods that we have seen during our research, which we consider relevant for the subsequent analysis of the methods we develop. The section is divided in three parts, which refer to methods from state of the art competitions, other approaches using competition datasets, and studies of unsupervised learning methods. For a deeper insight into the state of the art of the topic we recommend reading Margner's et al."Document analysis and text recognition" [31].

2.3.1 Methods on ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI)

Conferences intend to bring together international experts to share their experiences and to promote research and development. For this reason, we take a look at the best and worst performing methods of the last competition on writer identification [14], as well as some others we consider to be of interest. Note that we described the dataset it uses in the precious section. The best performing contestant, Tébessa II (Larbi Tebessi University, Department of Mathematics and Computer Science, Algeria) scores 76.4% TOP-1 55.6% mAP on the ICDAR2017 dataset. In this method, the different configurations of oriented Basic Image Features (oBIFs) columns histograms extracted from smoothed binary historical document samples with low-pass filters are concatenated for generating a feature vector and the City block distance measures is used for classifying each historical document.

We also considered interesting to remark the presence of the method Barcelona (Computer Vision Centre, Universitat Autònoma de Barcelona) which got a performance of 67.0% TOP-1 and 45.9% mAP. The method is totally learning free and uses grayscale images as input. Sparse Radial Sampling Local Binary Patterns (SRS-LBP) histograms at radii up to 12 are extracted for the full images and pooled globally to form an embedding of 3072. The features are then normalized and projected to 200 dimensions with a PCA transform.

Lastly, Fribourg (University of Fribourg, Switzerland and TU Kaiserslautern, Germany) gets the lower scores, of 47.8% TOP-1 and 30.7% mAP. The method uses a ResNet deep convolutional neural network (CNN), trained using the triplet margin loss metric to transform a given input into a space where inputs belonging to the same class (writer) are close to each other. The individual samples for the triplet consist of cropped (256$\times$256) sub-images from the input images.At testing

time, we generate a vector for each input image by averaging the embeddings produced by multiple random crops on the same input. Finally, the pairwise cosine distance between all input images are computed and the images are ordered in decreasing similarity to a given query image.

Method	**TOP-1**	**mAP**
Tébessa II	76.4	55.6
Barcelona	67.0	45.9
Fribourg	47.8	30.7

Table 2.2: Results from the ICDAR 2017 competition

2.3.2 Further state-of-the-art of methods

Recently published methods also use some of the benchmarking datasets which had been initially created for the competitions. This gives a fair comparison on the new approaches against other existing academic work. Here we describe some of those approaches.

The method YJ. Xiong et al. [45] propose is evaluated on te ICFHR2012-Latin and the ICDAR2013 datasets. For this approach, they present a method for text-independent writer identification using SIFT descriptor and contour-directional feature (CDF). The proposed method contains two stages. In the first stage, a codebook of local texture patterns is constructed by clustering a set of SIFT descriptors extracted from images. Using this codebook, the occurrence histograms are calculated to determine the similarities between different images. A candidate list of reference images is obtained for each image. The next stage is to refine the candidate list using the contour-directional feature and SIFT descriptor.

S. Fiel and R. Sablatnig present a method [16] evaluated on ICDAR2013 Competition on Writer Identification, and also ICDAR 2011 Writer Identification Contest, and the CVL-Database as the previous method. A feature vector is generated for each writer using Convolutional Neural Networks (CNN) and comparing them with previous known vectors. For the generation of those vectors, they cut off the output of the second last fully connected layer of a CNN trained on a database with known writers.

The last work of this kind we review is also trained on the ICDAR 2013 benchmark database. V. Christlein et al. The method [9] proposes the use of Zernike

moments evaluated at the contours of the script as local descriptor. Then a global descriptor is formed by encoding the extracted Zernike moments into Vectors of locally Aggregated Descriptors (VLAD). We will see how this approach obtains a great performance in Chapter 4

2.3.3 Unsupervised learning approaches

The previous examples have been concerned with supervised techniques, where label information of the authors are considered for training. Nevertheless, literature also shows an increasing interest on studying newer unsupervised methods resembling the purpose of our thesis.

Only one approach using autoencoders was found in the writer identification field. M. Elleuch et al. [12] highlight the effectiveness of Deep Learning techniques for recognizing Arabic handwritten script over the HACDB database [], and investigate two deep architectures: Deep Belief Network (DBN) and Convolutional Neural Networks (CNN). The experimental study has proved promising results which are comparable or even superior to the standard classifiers with an efficiency of DBN over CNN architecture. Nevertheless, the study focuses on an approach of writer identification in a character, instead of identifying complete handwritten documents, which is the goal of our methods.

On the same topic, V. Christlein, S. Fiel et al. [8] explore deep residual networks using surrogate classes, which as in [16] and [10] extract features from patches from activation layers of the network. We define this as our baseline, to be compared with the new approaches we propose. Further details are given in Chapter 3.

Chapter 3

Methodology

Chapter 3 contains an explanation of the methodology used for developing the project. Here we detail the entire process of writer identification from the very beginning to the end retrieval. The focus of the chapter mainly resides on the unsupervised feature extraction methods we propose.

"You can make anything by writing."

- C.S. Lewis

3.1 Introduction

As we discussed in Chapters 1 and 2, the methodology we present continues the work of V. Christlein et al. [8] mentioned previously, which we define as our baseline. That said, our methods propose new unsupervised learning alternatives to this approach based on the use of autoencoders. Figure 3.1 explains the main modules of the architecture of the methods we present. Algorithm workflow consists on: "Patch extraction", "Unsupervised feature learning" from the patches and "Encoding". The first module describes methods to obtain several small representations of the images we process. After that, characteristics of these patches are extracted by means of different algorithms, which are later encoded into an image descriptor in the last module, for proper writer recognition. The schematic will be used in section 3.5 to describe which submodules we consider to analyze by each method.

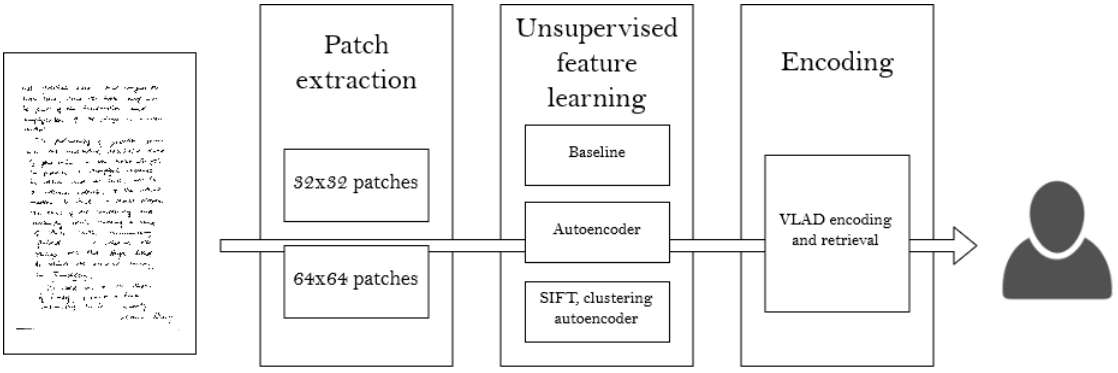


Figure 3.1: Modular schematic for our writer identification methods

The work of the thesis focuses on the first two modules, being "Encoding" beyond the scope of the project. The same encoding process is applied to both the original method and new approaches described in section 3.4. In addition, we consider preprocessing of the documents also out of the scope of the project. For this reason, provided pages are already binarized to ease the focus on the two modules "Patch extraction" and "Unsupervised feature learning". From those, we define three methods which aim to study and compare different unsupervised manners of extracting relevant information from the pages.

In the next sections we provide a detailed description of the concepts used on each of the modules, followed by an explanation of the new methodologies we studied for the thesis.

3.2 Patch extraction

A single handwritten page contains an amount of information which can be costly to process. For this reason, instead of processing an entire image, we take out small pieces of text as seen in Figure 3.2, and process them independently. This mechanism has been proven to be feasible for writer identification by S. Fiel and R. Sablatnig in [16], where patches are extracted from SIFT keypoint [29] locations. The description of the process is as follows.

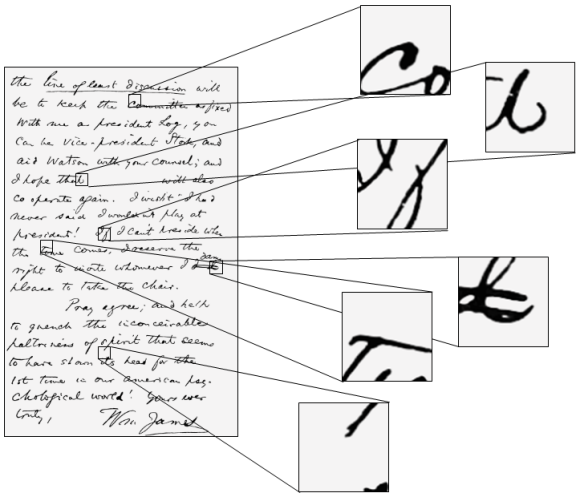


Figure 3.2: Extraction of random patches from a handwritten page

Keypoints on same or close locations are removed to avoid redundant characteristics, finally providing patches like the ones in Figure 3.3. From these elements, we intend to extract features for writer identification in section 3.3, and to encode information from pages for retrieval in section 3.4.

Figure 3.3: Patches from pages in the ICDAR 2017 dataset

3.3 Unsupervised feature learning

Machine learning methods can be classified into two different categories depending on whether we have access to label information or not. In supervised learning approaches we aim to predict the values of one or more output given a set of queries. Predictions are possible because of the training of these methods, which are based on cases solved for previous entries. These solutions for the inputs is what we call labels.

The methods we study and develop in this thesis, instead, address learning without target information in what we called an unsupervised way. What we aim is not to predict outputs anymore, but to estimate the probability distribution of the samples we have. Assuming our data lies on a low-dimensional manifold embedded in a higher-dimensional space, we expect to find ways to compress the information we have with low error. We recommend "Unsupervised Learning" by T. Hastie et al. [17] for deeper insights on the topic.

Regarding unsupervised learning methods, an autoencoder is a neural network that is trained to attempt to copy its input to its output. The structure of an autoencoder, seen in Figure 3.4, is composed of an encoder and a decoder, separated by a hidden layer in between. This layer is what later describes the code used to represent the input. Autoencoders have been successfully applied for dimensionality reduction and information retrieval tasks in the use of neural networks for unsupervised learning in P. Vincent et al. works [43] or [42], for instance. These facts, as well as the existence of successful examples on the MNIST dataset [23] (with data similar to ours), encourages the proposal of using these techniques for the problem we are addressing of unsupervised feature learning for writer identification, which has only been reported once in literature in M. Elleuch's [12] in a successful approach.

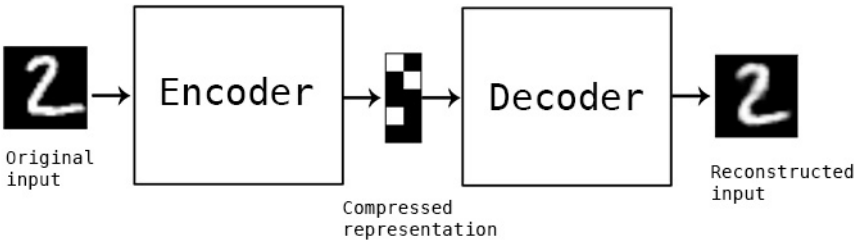


Figure 3.4: Autoencoder network basic structure. Image from [2]

We develop three different types of common autoencoders to better study the influence of each. To begin with, we construct a stacked autoencoder, consisting

of multiple layers in which the outputs of each layer is wired to the inputs of the successive layer like depicted in Figure 3.5. The layout structure presents variations from [6] inspired on the previous approach on autoencoders by M. Elleuch [12]. It is composed of and encoder and a decoder of three layers of size 1024 each (one more than in the Figure 3.5), using the ReLU (Rectified Lineau Unit) activation function, and a code size which is defined in the next chapter.

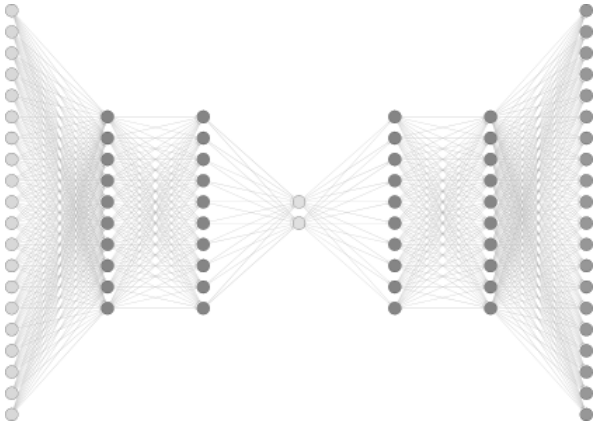


Figure 3.5: Stacked autoencoder network schematic. Image taken from [7]

The second architecture we propose is a convolutional autoencoder based on [4]. Convolutional autoencoders are a type of CNNs used for image dimensionality reduction, the structure of which resemble what we see in Figure 3.6. As we can observe, the built is based in an array of convolutional layers instead of fully-connected layers of the previous network. These networks have been used effectively in many machine learning tasks, achieving particular success in the domain of image processing in J. Jonathan Masci's et al. [32].

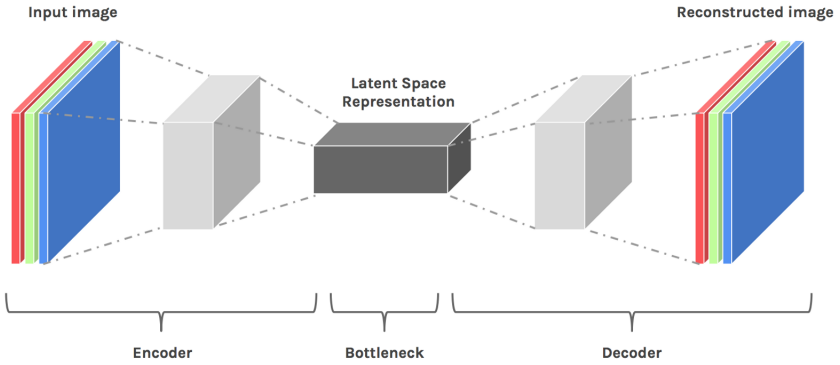


Figure 3.6: Convolutional autoencoder network schematic. Image taken from [1]

The last network we explore is a variational autoencoder. Following the example in Figure 3.7, P. Kingma et al. [20] consider a neural network with a probabilistic encoder, which does not output an encoding vector of size n, but rather outputs two vectors of size n: a vector of means, μ, and another vector of standard deviations, σ. Variational autoencoders have been used as generative models for instance in J. Walker et al. [44] defining their latent spaces, by design, continuous, allowing easy random sampling and interpolation. For this approach, we take the design from [5].

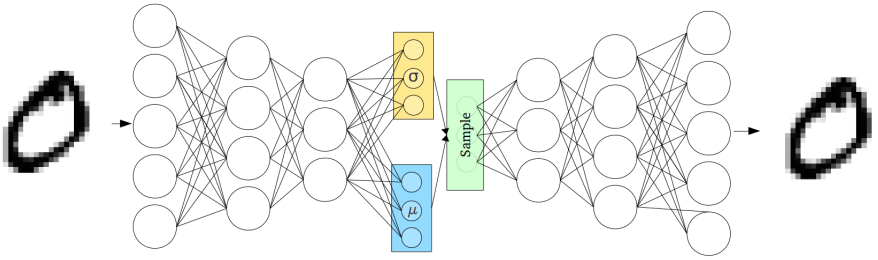


Figure 3.7: Variational autoencoder network schematic. Image taken from [3]

3.4 Encoding and retrieval

In this section we find a description of the last and common module depicted in the scheme from Figure 3.1. The aim of the encoding module is to aggregate all local patch descriptors into a single vector representation of the pages we analyze. This representations allow the use of metrics that define similarity between documents, prior to final identification and retrieval.

Following the method from the baseline, we encode by means of VLAD (Vector of Locally Aggregated Descriptors). For this solution, Jegou et al. [19] propose a descriptor derived from both BOF and Fisher kernel [34], which aggregates the set of local feature descriptors into a fixed-size vector that produces a compact representation of an image. The same as the BoVW (Bag of Visual Words) concept, a dictionary is the indispensable part in VLAD encoding. The idea of the VLAD coding is to generate a dictionary, which maps the local feature descriptors to the nearest codebook using K-means, with later normalization of the output. Figure 3.8 illustrates the methods of VLAD and a later approach of multi-Vlad encoding [25], both of which have proven successful in the baseline [8]. As a result, the use and comparison of both encoding methods are applied in our approaches.

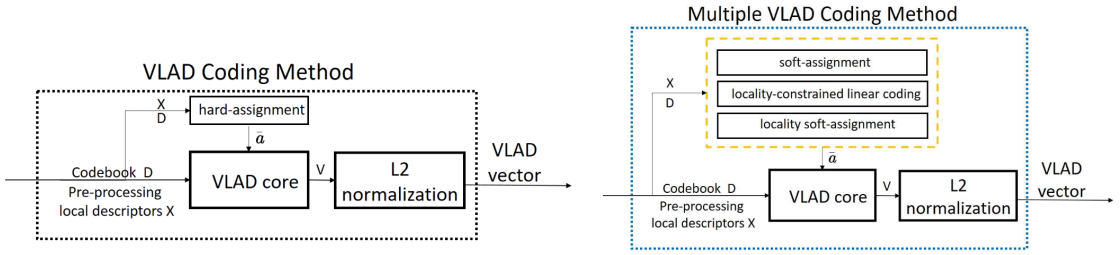


Figure 3.8: VLAD and m-VLAD encoding block diagrams. Image taken from [25]

3.5 Approaches for writer identification

In this section we illustrate the different approaches we have developed for the thesis. Starting with a replication of the original baseline, we describe the new methods we propose in feature extraction for writer identification. Results and analysis of the methods are displayed in Chapter 4.

3.5.1 Baseline

The baseline method describes the approach of V. Christlein et al. [8], the study and the replication of which were key to propose new methodologies to develop. Accordingly to Figure 3.9, the algorithm proposes a 32x32 patch extraction as explained in section 3.2.

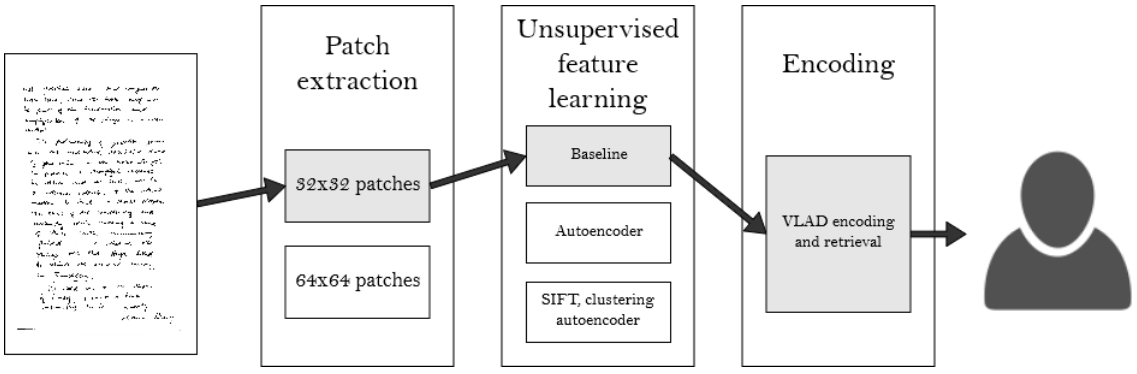


Figure 3.9: Baseline method [8] block scheme

Concerning the unsupervised feature learning module, the algorithm follows the structure shown in Figure 3.10. First of all, SIFT descriptors are computed as well at each keypoint location. Note that SIFT descriptors are invariant to both scale

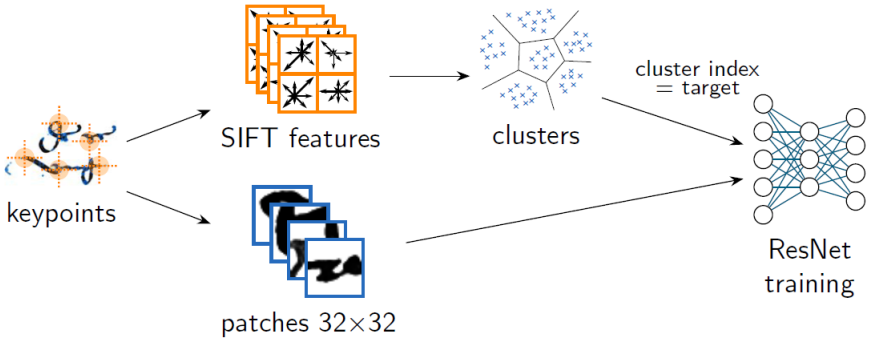


Figure 3.10: Schematics of the baseline feature extraction process. Image taken from [8]

and rotation changes, which explains the variance of the elements seen in Figure 3.11. The dimensionality of these descriptors is then reduced using principal component analysis (PCA) to lower the computational cost of the clustering process that is performed afterwards. From this information, we train a deep residual network (ResNet) [18] using patches as an input and cluster memberships as targets. Previous works also extract feature descriptors of patches from the penultimate layer of the networks [16][10]. Proper encoding and retrieval follows then, providing the method detailed previously in section 3.4.

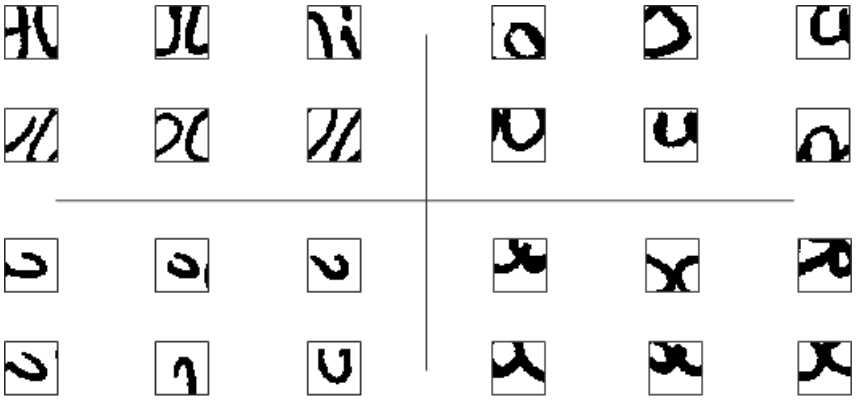


Figure 3.11: Example of patches from four different clusters extracted from the ICDAR 2013 dataset. Note that the SIFT descriptors are invariant to image scale and rotation and robustly match across distortion.

3.5.2 Vanilla approach using autoencoders

The first vanilla approach using autoencoders intends to analyze new unsupervised methods based on deep learning architectures. In contrast with the baseline, this method explores a pure deep learning-based solution.

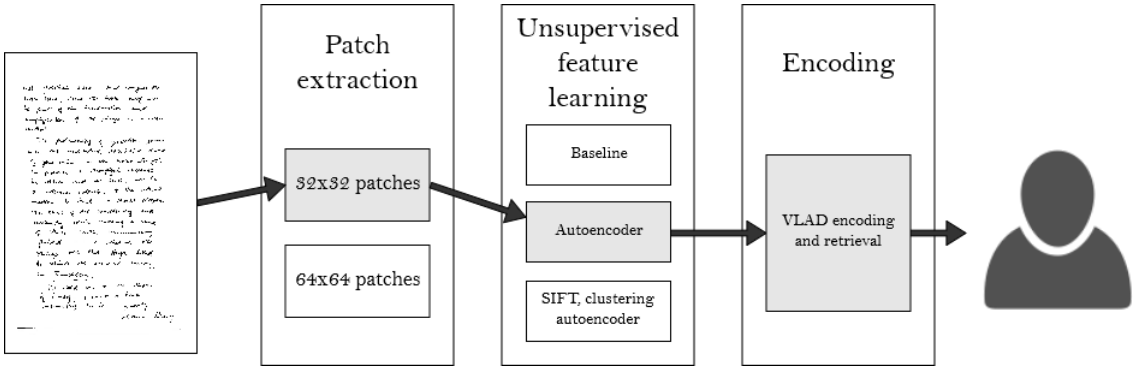


Figure 3.12: Experiment 2 block scheme

From Figure 3.12 we can see that the method uses the same patch extraction than 3.5.1. Regarding unsupervised learning, no clustering membership information is provided, as we do not compute SIFT descriptor. Instead, we train an autoencoder to learn characteristics from the data by encoding them into a lower sized layer, which is then decoded to try to replicate the input. The feature vectors are obtained by forwarding the patches through the trained encoder, taking information from the intermediate layer of an autoencoder. Results and comparison of architectures described in 3.3 are shown in section 4.3 of Chapter 4.

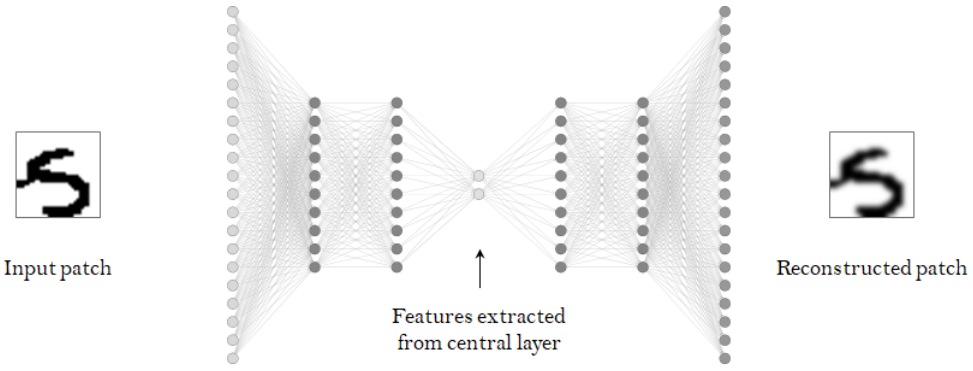


Figure 3.13: Experiment 2 graphic description

3.5.3 Increase of patch size

For the second methodology applied we modify the patch extraction block. Figure 3.14 indicates the decision in which we process 64x64 patches instead of the previous 32x32 ones, in order to evaluate the impact of the patch extraction module. The unsupervised learning block continues to use autoencoders, which maintain the same architecture introducing modifications only to adjust new input and output sizes.

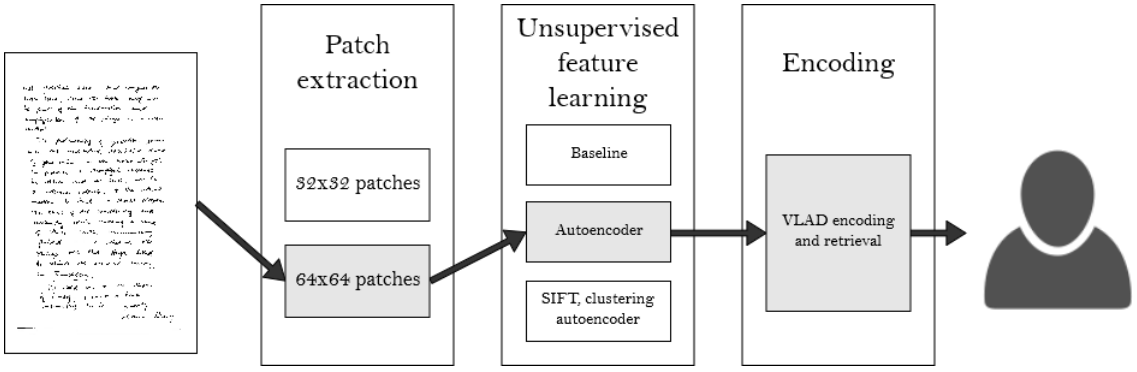


Figure 3.14: Experiment 3 block scheme

The increase of the patches' size can be appreciated in Figure 3.15. New images to encode are 4 times larger, which can contain several characters and even entire words (see Figure 3.16) where SIFT keypoints locations are. By having access to that information, we expect an increase of performance in writer identification, at least on identifying pages on the same language than the query. On the other hand, a change on the alphabet could also penalize the identification, as patch handwriting can be interpreted from different writers because of that. Results and analysis are shown in section 4.4.

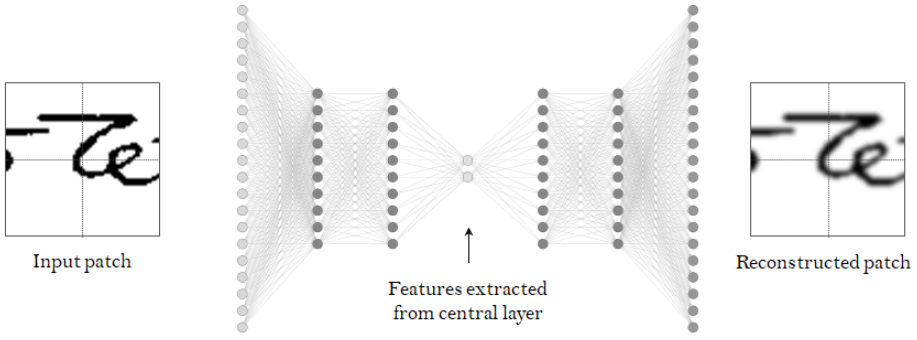


Figure 3.15: Experiment 3 graphic description

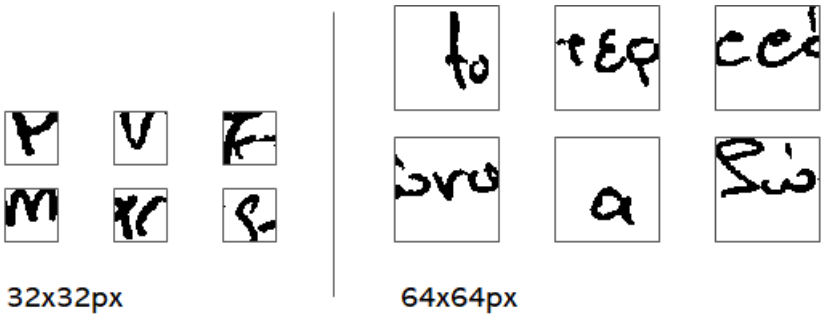


Figure 3.16: Difference between 32x32px and 64x64px patches

3.5.4 Using SIFT descriptor information for autoencoders

As seen in Figure 3.17, the last method we propose explores a new approach for the unsupervised feature learning block. Unlike in the previous cases 3.5.2 and 3.5.3, we exploit the information of cluster membership of the SIFT descriptors like in the baseline method 3.5.1. Therefore, this experiment tries to integrate autoencoder methods as well as the original one. From this approach we expect the autoencoders to learn specific treats from writers which could not have been acquired by simply reconstructing the same image.

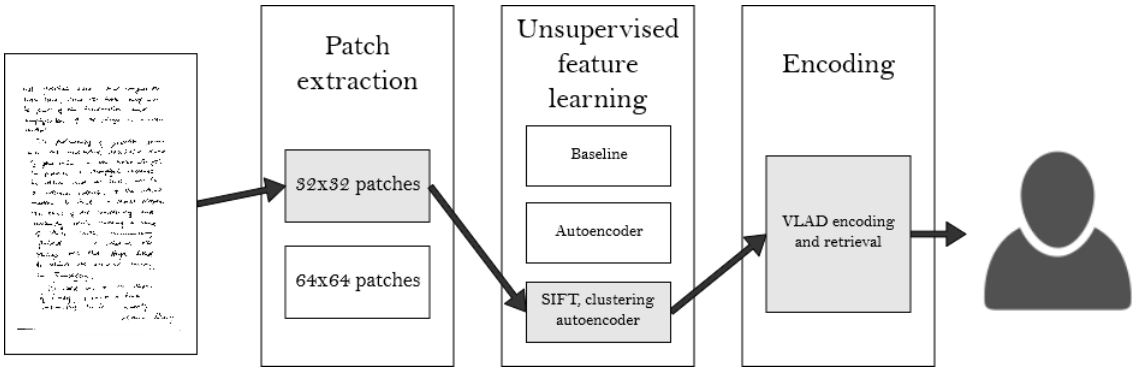


Figure 3.17: Experiment 4 block scheme

The algorithm considers the scheme seen in Figure 3.18. First, we compute the SIFT descriptors from each patch and apply clustering as in the baseline. After that, patches which share cluster and page membership are grouped prior to

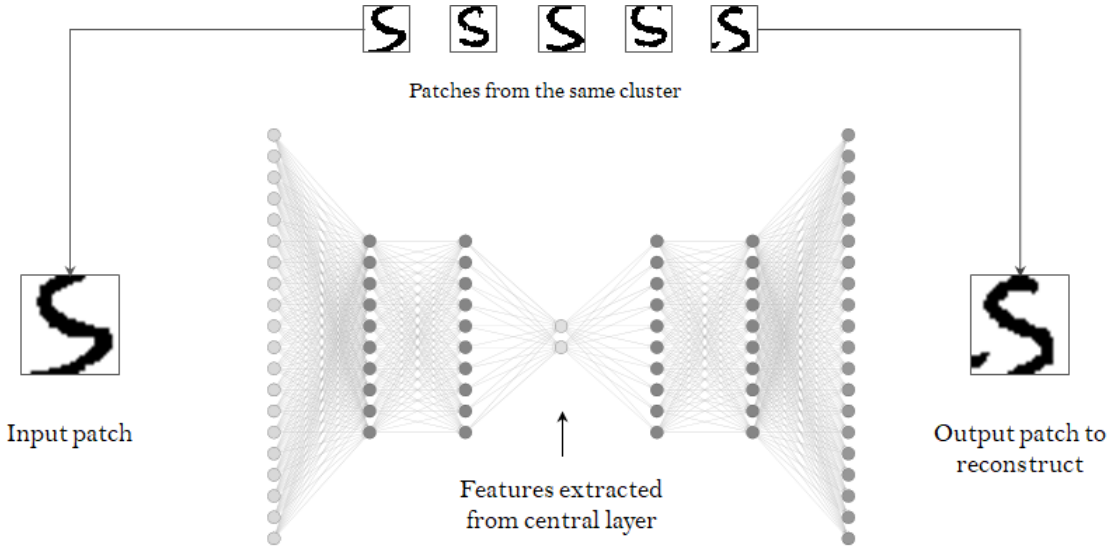


Figure 3.18: Experiment 4 graphic description

training of the autoencoder. The main distinction of the method is that the autoencoder does not try to reconstruct the patches, but a random patch from the same group, examples of which are shown in Figure 3.19. The discussion and the results of this method are found in section 4.5, where we will see if this strategy provides improvements in extraction of specific features from writers as intended.

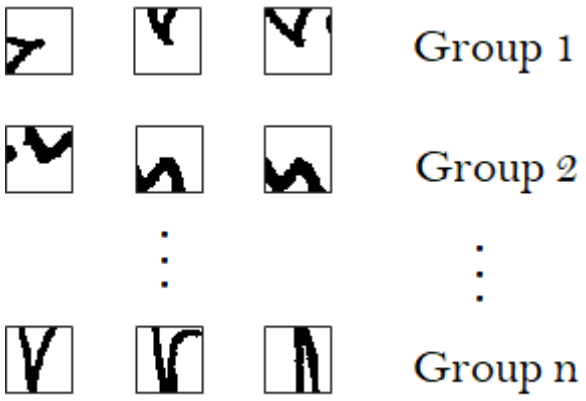


Figure 3.19: Examples of groups of patches which share page and cluster membership

3.6 Computer Vision Lab environment

In this section we comment briefly implementation details of the realization of the project. The code for the experiments can be found in [1] which was forked from the original [2]. Table 3.1 below shows the available hardware and software resources for the development of the thesis.

Hardware	
CPU	Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz
Memory	16 GB
GPU	Nvidia GeForce(R) GTX 980

Software	
OS	Ubuntu 16.0.4
Python version	Python 3.6.4 :: Anaconda, Inc.
Deep Learning Framework	PyTorch 0.3.1

Table 3.1: Remote hardware and software resources

[1] https://smithers.cvl.tuwien.ac.at/pallas/tf-compare
[2] https://smithers.cvl.tuwien.ac.at/tensorflow/cvl-pytorch

Chapter 4

Results and analysis

In this chapter we show the results obtained from the methods described in the previous chapter, evaluated using the ICDAR2013 and ICDAR2017 datasets. First in the chapter, we detail the evaluation metrics we use for comparison. Next sections explain the experiments we performed and provide a detailed analysis. The last section of the chapter makes a summary of the best methods, and compares it with the baseline and state of the art.

"If you tell the truth,
you don't have to remember anything."

- Mark Twain

4.1 Evaluation metrics

The main objective of the thesis is to evaluate and compare the algorithms we explored for the task of writer identification and retrieval. For this reason, excluding "Precision from patch", we define common metrics in writer identification competitions [26] [14] following:

Precision from patch: It is the simplest evaluation we perform on writer identification, only mentioned in the first experiments for network training purposes. Given a random sample of patches from pages of the test dataset, we compute the probability of successful identification provided one single patch.

Hard and soft TOP-N, and TOP-1 : In order to measure the accuracy of a writer identification method, we use the soft TOP-N and hard TOP-N criteria used in [26] and [14]. For all documents in the dataset we calculate the distance to all other documents and sort them by similarity (distance score) to the query using leave-one-out methodology. For all document images of a benchmarking dataset, we count the correct hits. The ratio of the total number of correct hits to the total number of the document images in the benchmarking dataset corresponds to the TOP-N accuracy.

For the soft TOP-N criterion, we consider a correct hit when at least one document image of the same writer is included in the N most similar document images. Concerning the hard TOP-N criterion, we consider a correct hit when all N most similar document images are written by the same writer. Note that the maximum value of N for the hard criterion is related to the total number of documents that were written by a writer, as we cannot retrieve more documents from one writer than there are in the dataset

Note that the measure is equivalent for both hard-1 and soft-1, defining TOP-1. This particular metric describes the probability that the document ranked first on retrieval corresponds to the same author than the query image. We use this metric as the evaluation of the accuracy of writer identification.

Mean Average Precision (mAP) is defined over the Precision at N (p@n) criterion, which calculates the percentage of hits within the first N ranked documents. Mean Average Precision computes the average of p@N calculated over all N values. The purpose of this metric is to evaluate if the ranking of retrieval of the pages. Using the definition from the ICDAR2017 Competi-

tion on Historical Document Writer Identification (Historical-WI) [14], mAP is calculated as follows:

$$mAP = \frac{\sum_{q \in Q} AveP(q)}{Q}$$

where Q is the set of all documents and q the current query document image, and $AveP$ the corresponding average precision. The average precision is the area under the precision-recall curve and also takes the position of the positive samples in the ranking into account. We define it as:

$$AveP = \frac{\sum_{k \in n} P(k) \times rel(k)}{numberofrelevantdocuments}$$

where P is the precision, *rel(k)* is an indicator function equaling 1 if the item at rank k is a relevant document, zero otherwise, and n is the number of all documents in the dataset. We showcase some examples of mAP in Figure 4.1 to facilitate the understanding of the metric.

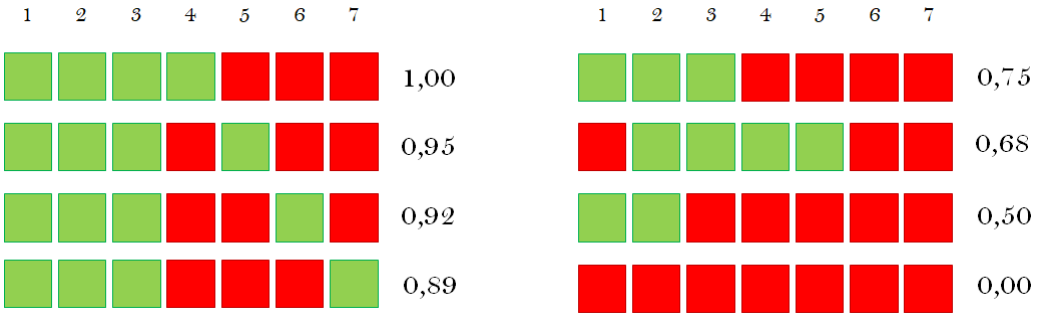


Figure 4.1: Example of calculations of mAP for writer retrieval. The example shows a scenario where there are 4 other pages of same authorship than query (in green) over a dataset of 33 pages. Ranking of the documents is organized from left to right following the numeration on top. The Figure illustrates the effects of different ranking scenarios to better understand the metric

4.2 Feature vector size evaluation

In this first experiment we aim to empirically evaluate which feature size works best for the autoencoders we described in section 3.3. The optimal sizes will be later used for the writer identification and retrieval methods in all of the experiments following. For this reason, we define a quantitative (Table 4.1) and qualitative (Figure 4.5) analysis of the results to determine the influence of using 8, 16, 32, 64 and 128 numbers of features per patch.

The reconstruction loss of the autoencoders we propose is the well-known Mean Squared Error (MSE), which is defined as: $MSE = \frac{1}{N}\sum(f_i - \hat{f}_i)^2$ where N is the number of samples and $\hat{f}_i$ is our estimation of f_i. Both qualitative and quantitative results are collected after 20 epochs over the ICDAR 2013 dataset.

Architecture	**8**	**16**	**32**	**64**	**128**
Stacked autoencoder (SAE)	0.062	0.040	0.029	0.027	0.025
Convolutional autoencoder (CAE)	0.096	0.082	0.049	0.041	0.038
Variational autoencoder (VAE)	0.105	0.082	0.056	0.042	0.041

Table 4.1: MSE in reconstructions after 20 epochs

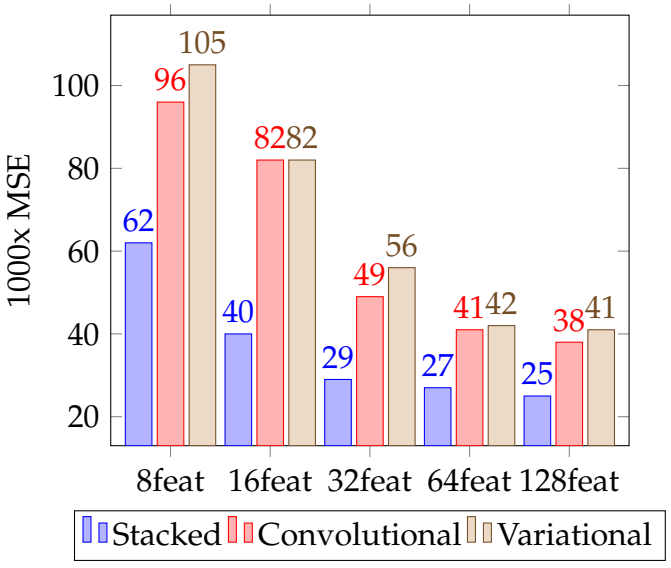


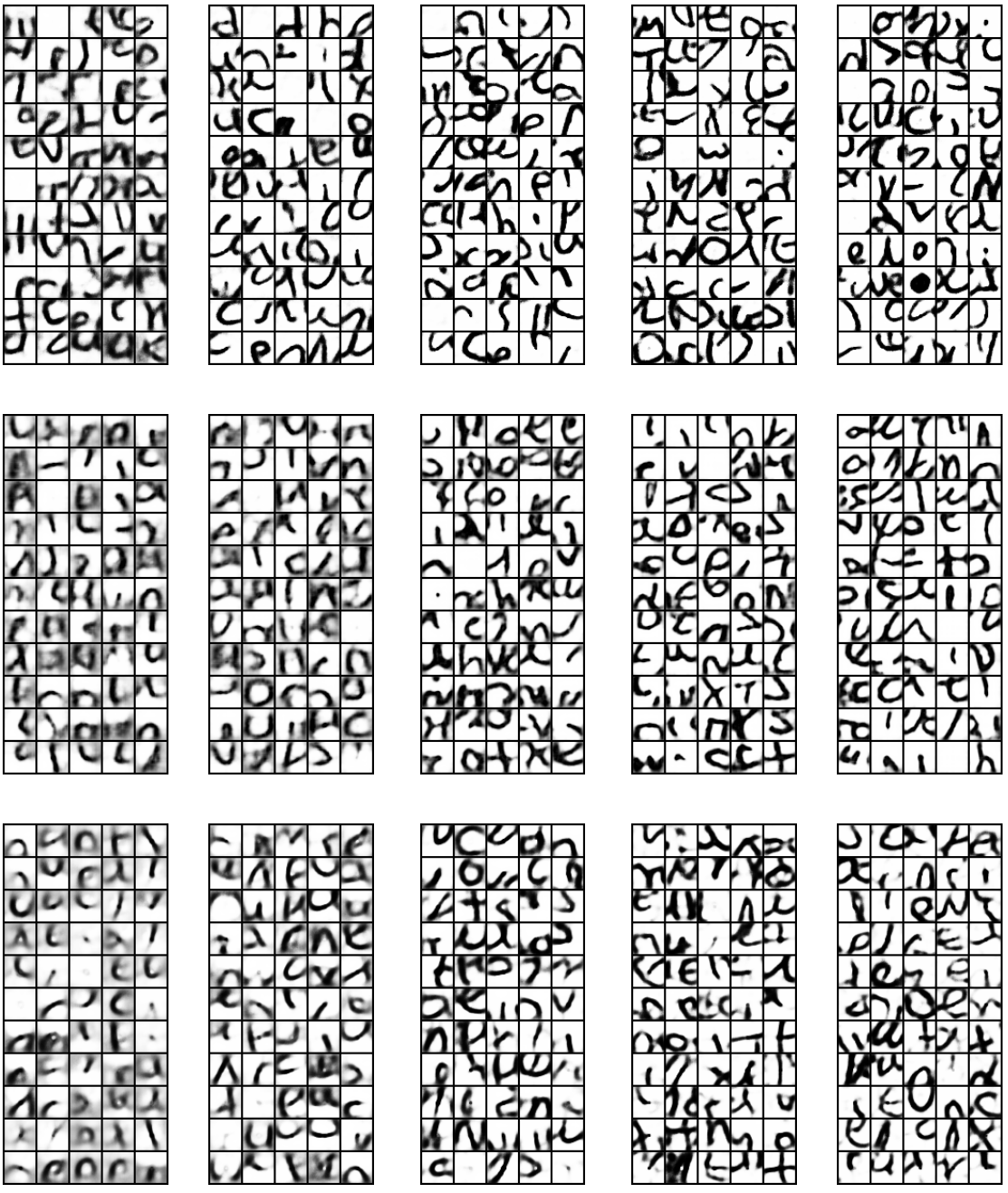
Figure 4.2: MSE in reconstruction comparison

Influence of size of feature vector

As we could have expected, an increase of the central layer size results into a better reconstruction loss of the patches because the least we compress the data, the better the reconstruction is, as Figure 4.2 shows. Nevertheless, we do not see significant improvement from feature sizes of 64 and 128 for any of the autoencoders. Considering the trade off between feature size and reconstruction loss, we consider a smaller feature size could also generalize better. Consequently, 64 looks like the adequate feature number for next experiments in all architectures.

Concerning to each architecture we can state that:

1. SAE achieves a remarkably low reconstruction loss even with a very low number of features, but does not improve as much for higher values.
2. CAE gets a higher loss than the SAE both qualitative and quantitatively for the same number of features.
3. VAE obtains similar quantitative results than CAE, but resulting into a slightly better qualitative reconstruction, still not comparable with SAE.



(a) 8 features (b) 16 features (c) 32 features (d) 64 features (e) 128 features

Figure 4.5: From top to bottom, patch reconstructions of stacked, convolutional and variational autoencoders by feature vector size for qualitative analysis

4.3 Vanilla approach with autoencoders

This section evaluates the method described in section 3.5.2 and compares the performance of the different architectures of autoencoders from 3.3. Precision, TOP-1 and mAP results from Table 4.2 are collected from the ICDAR 2013 and ICDAR 2017 test datasets, after 30 epochs of training in both train datasets respectively. We apply the two encoding methods we described in section 3.4 to test if we can see any improvements using multi VLAD encoding (m-VLAD) against single VLAD, with a codebook size of 30000. For the m-VLAD approach we use VLAD sizes of 51, 25, 12, 6 and 3.

Because of non-convergent oscillations in training, we estimated error measures of $\pm 1\%$. In order to reduce spurious values, we select the median of the three last epochs which reduce error uncertainty to $\pm 0.5\%$

Architecture	**ICDAR 2013**			**ICDAR 2017**		
	Prec	**TOP-1**	**mAP**	**Prec**	**TOP-1**	**mAP**
SAE VLAD	5.89	85.9	59.1	1.94	61.3	42.3
CAE VLAD	4.98	83.8	59.0	1.72	58.6	39.8
VAE VLAD	5.21	83.6	59.1	2.01	60.1	41.6
SAE m-VLAD	-	87.1	63.0	-	61.9	40.9
CAE m-VLAD	-	86.0	61.1	-	59.8	39.1
VAE m-VLAD	-	86.3	60.0	-	61.8	41.1

Table 4.2: Results for the Experiments 1 and 2

Influence of unsupervised learning architecture

Although we do not see relevant differences among the results in both TOP-1 and mAP, we observe a slightly better performance for SAE, observing a very similar behavior for CAE and VAE in both datasets and encoding methods. Nevertheless, differences do not seem clear enough to define the best architecture.

Regarding Precision from patch (Prec), identifying a writer from a single patch shows very poor results. Nevertheless, values surpassing random guesses are sufficient (1.5% for ICDAR 2013 and 0.53% for ICDAR 2017), which helps us know the algorithms are actually learning.

Influence of encoding

Multi VLAD encoding had already been proven to boost encoding and retrieval performance in [8] and [19]. In the case of our architectures we observe performances of up to a 2.2% in TOP-1 and 4% in mAP in the ICDAR 2013, with slight improvements on the ICDAR 2017 dataset too. For this reason, we will use m-VLAD encoding for next experiments to exploit the best possible performance.

Influence of dataset

A clear difference can be observed for the dataset performance results compared. Main reasons are: size of the dataset (number of writers, and pages as well), and simplicity of the samples. As we can see in Figure 4.6, ICDAR 2017 pages turn out to be more challenging to process, as we an observe annotations of other writers, or more difficult layouts than for ICDAR 2013.

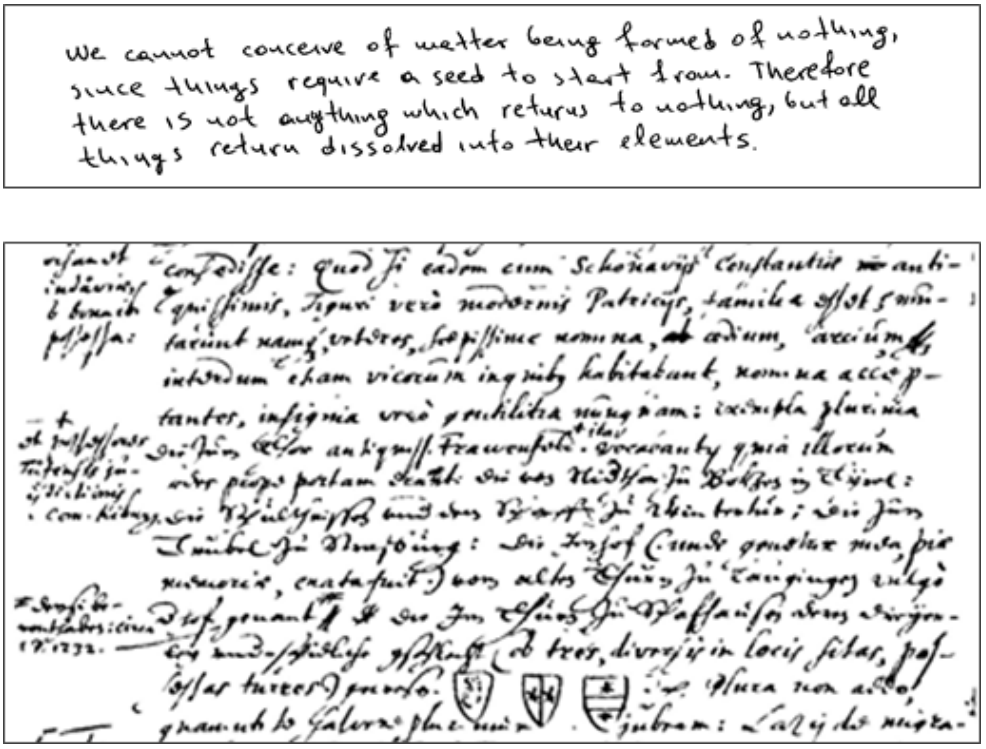


Figure 4.6: Comparison of pages from ICDAR 2013 (top) and ICDAR 2017 (bottom)

4.4 Increase of patch size

After the implementation of the second method we propose, we want to conclude whether an increase on the patch size provides more significant characteristics, resulting on an increase in recognition or retrieval performance. The following section evaluates the influence of the extraction of 64x64 patches instead of 32x32, proposed in section 3.5.3.

Autoencoders are trained on the ICDAR 2013 train dataset and evaluated on its test one. Fine tuning learning parameters is done to achieve the best performance, although the core architectures remain intact introducing small modifications to fit the new input and output sizes. Furthermore, we extract the same number of patches per page (1000) so that we make sure more information than in Experiment 3.5.2 is provided. Like in the previous section, results are collected after 30 epochs using the same methodology. In Table 4.3 we see the results obtained for TOP-1 and mAP against the ICDAR 2013 dataset, with the addition of hard and soft top-N metrics.

Architecture	**TOP-1**	**Hard**		**Soft**		**mAP**
		top-2	**top-3**	**top-2**	**top-5**	
SAE 32x32 m-VLAD	87.1	40.8	20.9	91.7	96.4	63.0
SAE 64x64 m-VLAD	85.6	33.5	17.7	89.8	94.6	57.7
CAE 32x32 m-VLAD	86.0	37.8	18.1	92.2	95.3	61.1
CAE 64x64 m-VLAD	83.1	31.6	15.2	87.5	92.9	55.4
VAE 32x32 m-VLAD	86.3	35.2	18.7	91.0	96.0	60.0
VAE 64x64 m-VLAD	90.2	42.1	22.2	94.7	96.9	64.2

Table 4.3: Results for 64x64 patches on ICDAR 2013 with m-VLAD encoding

Influence of architectures and patch size

VAE is the only method which performs better with increased size performing the best in all metrics. The network seems to better estimate the probability density function when gathering more input information, which leads to better results.

The decrease of performance could be explained in CAE, as larger patches are downsampled for this network, using average pooling in other to modify the network the least. However, from this result we can extract interesting conclusions. We could think that the increase of information per patches is be given because of an increase of characters per image as seen in Figure 4.7. Nevertheless, in the case of CAE we prove that more characters per patch do not provide more information about the writer.

We can also confirm that SAE and CAE perform clearly worse, specially in the retrieval task. We observe a great decrease in mAP, hard top-3 and specially hard top-2. However, differences are not big enough for identification in TOP-1. A reason which could explain that is that the algorithm may be alphabet dependent. Given a query image, it finds the other one in the same language (high TOP-1), but fails to identify the same writer ones in another language (low hard top-2). This statement could be valid considering that 32x32 patches use to get parts from characters, whereas in 64x64 we can clearly differentiate the alphabet of the pages as we can observe in Figure 4.7.

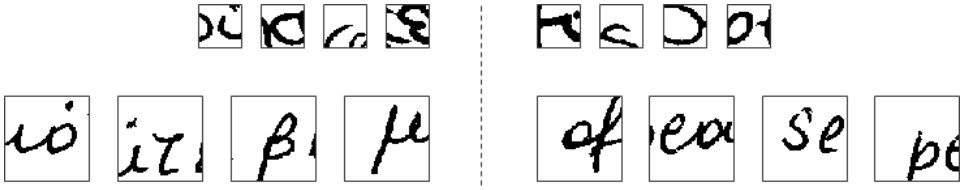


Figure 4.7: Difference between 32x32px and 64x64px patches in two alphabets. Note in the small patches it is very difficult or almost impossible to tell the difference of alphabets in most cases, in contrast to larger patches.

4.5 Using SIFT descriptor information for autoencoders

The last experiment we conduct is based on the method from section 3.5.4. In this case we want to evaluate the viability of the method in which the algorithm does not rely on just to reconstruct images anymore, but random patches sharing page and cluster membership. This is done to extract specific characteristics from writers we believe could not have identified any other way. Once again, we evaluate the method over ICDAR 2013 using the same methodology than in sections 4.3 and 4.4. Table 4.4 displays previous results too for easier comparison.

Architecture	**TOP-1**	**Hard**		**Soft**		**mAP**
		top-2	**top-3**	**top-2**	**top-5**	
SAE m-VLAD	87.1	40.8	20.9	91.7	96.4	63.0
SAE 64x64 m-VLAD	85.6	33.5	17.7	89.8	94.6	57.7
SAE Cl m-VLAD	89.0	40.6	20.6	92.8	96.2	63.0
CAE m-VLAD	86.0	37.8	18.1	92.2	95.3	61.1
CAE 64x64 m-VLAD	83.1	31.6	15.2	87.5	92.9	55.4
CAE Cl m-VLAD	80.8	36.4	18.2	86.7	93.4	58.5
VAE m-VLAD	86.3	35.2	18.7	91.0	96.0	60.0
VAE 64x64 m-VLAD	90.2	42.1	22.2	94.7	96.9	64.2
VAE Cl m-VLAD	88.3	40.5	22.8	92.4	95.8	63.0

Table 4.4: Results for patches using clustering information (Cl) on ICDAR 2013

Influence of SIFT clustering membership information

Providing this new approach SAE experiences a small boost performance in identification, increasing TOP-1 by a not so relevant 2%. In the case of CAE, however, we achieve the worst TOP-1 from all methods, including a decrease in all other metrics respect the vanilla method (except hard top-3, remaining the same). As for VAE, an interesting increase of performance is achieved in all metrics from the vanilla approach. VAEs are generative networks, applied to generate a random, new output, that looks similar to the training data. This reason could explain the boost when used to generate new data. Nevertheless, performance is still worse than *VAE 64x64 m-VLAD*, which we consider to be the best approach of all we tried.

4.6 Results summary

In the last section of the chapter we evaluate our methods with state-of-the-art solutions over ICDAR 2013 and ICDAR 2017 datasets. The analysis are first made with the baseline method [8] to determine the improvement or worsening of the new proposals. After that, we compare with a wider range of algorithms that we consider interesting to have a larger overview of performance. In Table 4.5 we contrast the best result we obtained over the ICDAR 2013 dataset with the baseline we replicated. As it can be seen, our method performs worse than the baseline in all metrics, seeing a bigger difference in hard top-2.

Architecture	**TOP-1**	**Hard**		**Soft**		**mAP**
		top-2	**top-3**	**top-2**	**top-5**	
Baseline	93.1	51.2	27.4	96.6	98.4	69.7
VAE 64x64 m-VLAD	90.2	42.1	22.2	94.7	96.9	64.2

Table 4.5: Best results from all experiments on ICDAR 2013

The experiment is also replicated, only for the vanilla approaches, in the ICDAR 2017 dataset, where the baseline proved to achieve outstanding performance in [8]. As we can see in Table 4.6, this time, our methods performance is extremely lower in all cases. A possible hypothesis is that the new dataset is much larger, meaning networks may be too small for it.

Architecture	**TOP-1**	**Hard**		**Soft**		**mAP**
		top-2	**top-3**	**top-2**	**top-5**	
Baseline	93.1	51.2	27.4	96.6	98.4	69.7
SAE m-VLAD	61.9	38.0	21.6	66.6	73.1	40.9
CAE m-VLAD	59.9	36.0	19.8	65.8	72.2	39.1
VAE m-VLAD	61.8	37.2	21.4	67.2	73.8	41.1

Table 4.6: Best results from all experiments on ICDAR 2017

In the ICDAR 2013 dataset we proved an increase of performance of the non-vanilla autoencoder methods which could also be implemented on the current dataset. However, we do not expect an improvement significant enough to be compared with the baseline. Still, results need to be collected to prove it.

On a larger scale, we compare baseline and our methods in Table 4.7. Our method performs slightly better than S. Fiel and R. Sablatnig's [16], but it is clearly outperformed by the rest.

Architecture	TOP-1	Hard		Soft		mAP
		top-2	top-3	top-2	top-5	
Baseline	93.1	51.2	27.4	96.6	98.4	69.7
VAE 64x64 m-VLAD	90.2	42.1	22.2	94.7	96.9	64.2
Xiong et al. [45]	96.2	63.5	35.0	-	98.6	-
Fiel and Sablatnig [16]	88.5	40.5	23.8	92.2	96.0	-
Christlein et al. [9]	99.4	81.0	61.8	-	-	-

Table 4.7: Best results from all experiments on ICDAR 2013

Even more comparisons were performed over the ICDAR 2017 dataset, comparing to contestants for the Competition on Historical Document Writer Identification (Historical-WI) [14] in Table 4.8.

Architecture	TOP-1	Hard		Soft		mAP
		top-2	top-3	top-2	top-5	
Baseline	88.6	77.1	64.7	-	92.2	74.4
SAE m-VLAD	61.9	38.0	21.6	66.6	73.1	40.9
CAE m-VLAD	59.9	36.0	19.8	65.8	72.2	39.1
VAE m-VLAD	61.8	37.2	21.4	67.2	73.8	41.1
Tébessa II	76.4	56.6	37.8	-	86.6	55.6
Barcelona	67.0	45.1	27.4	-	76.9	45.9
Fribourg	47.8	24.7	12.6	-	62.1	30.7

Table 4.8: Best results from all experiments on ICDAR 2017

Comparing to all competing methods (we add Hamburg, Groningen and Tébessa I, which were not mentioned before), Figure 4.8 shows how we position ourselves in a low rank position for writer identification (TOP-1), only outperforming Fribourg proposal and close to the Barcelona approach. On the other hand, the baseline obtains the best results, remarkably better than the second best approach.

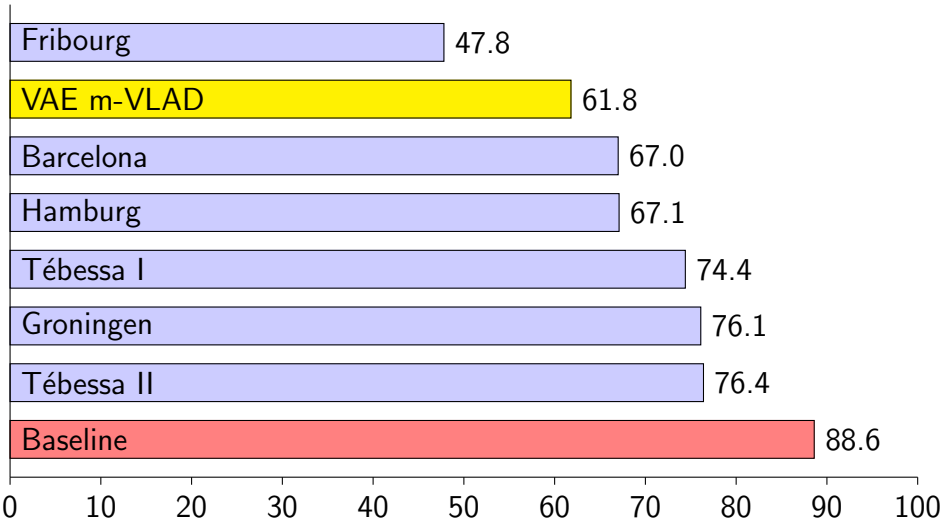


Figure 4.8: TOP-1 comparison graph with results from ICDAR 2017

Concerning writer retrieval (mAP) in Figure 4.9, results remain similar than before. We can say then, that we prove to still be far from the baseline approach and best state-of-the-art methodologies. As a result, we believe the use our autoencoder-based methods do not provide an improvement in writer identification and retrieval feature extraction solutions for handwritten documents.

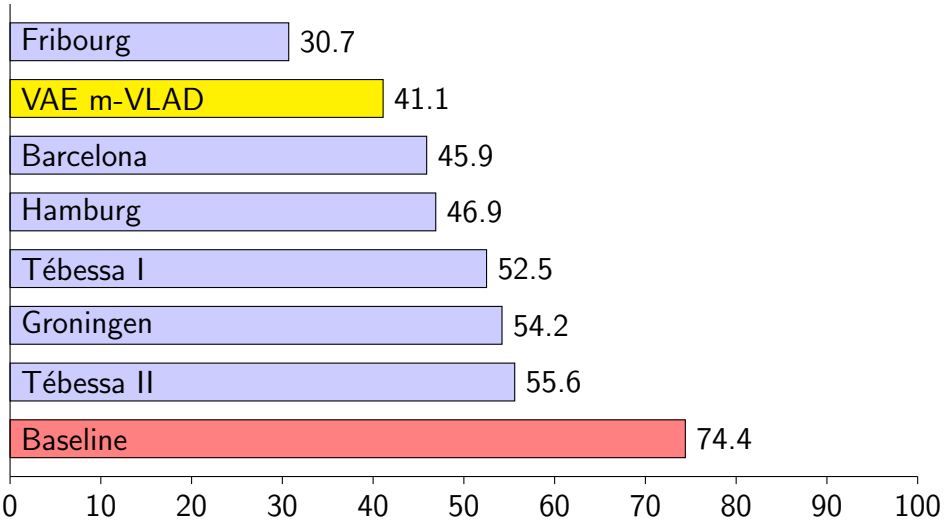


Figure 4.9: mAP comparison graph with results from ICDAR 2017

Chapter 5

Conclusions

The last chapter of the document is divided in two sections. In the first one we include the achievements of the thesis, regarding the objectives proposed at the beginning. Following, we propose recommendations for future work.

"Don't adventures ever have an end?
I suppose not.
Someone else always has to carry on the story."

- J.R.R. Tolkien, The Fellowship of the Ring

5.1 Achievements

In accordance with the objectives we defined at the beginning of the thesis, I believe I have been able to learn and develop skills on deep learning programming. After the development of the thesis, I was capable to build up deep learning algorithms and understand the concepts behind the implementation. Furthermore, I have learned about state of the art methods in writer identification and retrieval, with which the replication of the state of the art baseline contributed to achieve deeper insights on the topic.

Concerning the main objective of the thesis, we successfully developed unsupervised methods of feature extraction for writer identification based on neural networks. In the experimental section 4.6 we showed that we were able to compare our approaches based on autoencoders with state of the art methods in benchmarking datasets. With our best method, we managed to achieve a 90.2% TOP-1 and 64.1% mAP against the ICDAR 13 dataset, and 61.8% TOP-1 and 41.1% mAP against ICDAR 2017. Comparing with the baseline method however, we observe a decrease in performance, specially on the second dataset. The exploration of these methods conclude that, up to date, our use of autoencoders in writer identification is not able to improve state of the art methods of feature extraction yet. Taking into consideration the topic is still not common in literature, we believe that our thesis opens a new field to explore.

5.2 Future work

Future research should consider the potential effects of the increase of patch size and clustering membership information more carefully. For example, we consider interesting to see the effects a method which performed a merging of the two approaches. In addition, these approaches could be replicated on the ICDAR 2017 dataset to observe the differences compared to the vanilla autoencoder results.

Other new lines of research would lead to develop more deep and complex networks. As a procedure to explore, it could be implemented an autoencoder whose encoder was based on the baseline network, building a mirrored symmetric architecture for decoder.

Chapter 6

Budget

For the cost of the student budget it has been taking into account a salary of 10 €/hr. Considering 1 ECTS is equal to 25 study hours, for the 30 ECTS Master's Thesis we estimate a workload of 750€.

Average supervisor assessment cost is set to 30 €/week, with 22 weeks for the duration of the stay.

The software uses open source licenses for its development. While hardware has been provided by the TU Wien at not additional charge.

Total cost adds a 21 % additional budget overhead because the project has been developed within the Universitat Politècnica de Catalunya (UPC) study plan.

Student cost	10 €/hr x 750 hr	7,500 €
Supervisors cost	30 €/week x 22 weeks	660 €
Hardware and software resources		0 €
Subtotal		8,160 €
UPC overhead	21% of the Subtotal	1,713 €
TOTAL		**9,873 €**

Table 6.1: Budget of the Master's Thesis

Bibliography

[1] A Theory of Vibe — Glass Bead.
http://www.glass-bead.org/article/a-theory-of-vibe/?lang=envie.

[2] Building Autoencoders in Keras.
https://blog.keras.io/building-autoencoders-in-keras.html.

[3] Intuitively Understanding Variational Autoencoders – Towards Data Science.
https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf.

[4] L1aoXingyu/pytorch-beginner · GitHub.
https://github.com/SherlockLiao/pytorch-beginner/blob/master/08-AutoEncoder/conv_autoencoder.py.

[5] L1aoXingyu/pytorch-beginner · GitHub.
https://github.com/SherlockLiao/pytorch-beginner/blob/master/08-AutoEncoder/Variational_autoencoder.py.

[6] MorvanZhou/PyTorch-Tutorial · GitHub.
https://github.com/MorvanZhou/PyTorch-Tutorial/blob/master/tutorial-contents/404_autoencoder.py.

[7] Simple MNIST Autoencoder in TensorFlow · Gertjan van den Burg.
https://gertjanvandenburg.com/blog/autoencoder/.

[8] V. Christlein, M. Gropp, S. Fiel, and A. Maier. Unsupervised Feature Learning for Writer Identification and Writer Retrieval. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 991–997, Nov 2017.

[9] Vincent Christlein, David Bernecker, and Elli Angelopoulou. Writer identification using vlad encoded contour-zernike moments. In 2015 13th

International Conference on Document Analysis and Recognition (ICDAR), pages 906–910. IEEE, 2015.

[10] Vincent Christlein, David Bernecker, Andreas Maier, and Elli Angelopoulou. Offline writer identification using convolutional neural network activation features. In German Conference on Pattern Recognition, pages 540–552. Springer, 2015.

[11] C. Clausner, A. Antonacopoulos, and S. Pletschacher. ICDAR2017 Competition on Recognition of Documents with Complex Layouts - RDCL2017. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 1404–1410, Nov 2017.

[12] Mohamed Elleuch, Najiba Tagougui, and Monji Kherallah. Towards unsupervised learning for Arabic handwritten recognition using deep architectures. In International Conference on Neural Information Processing, pages 363–372. Springer, 2015.

[13] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2963–2970. IEEE, 2010.

[14] S. Fiel, F. Kleber, M. Diem, V. Christlein, G. Louloudis, S. Nikos, and B. Gatos. ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI). In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 1377–1382, Nov 2017.

[15] Stefan Fiel. Novel methods for writer identification and retrieval. PhD thesis, Technische Universität Wien, 2016.

[16] Stefan Fiel and Robert Sablatnig. Writer identification and retrieval using a convolutional neural network. In International Conference on Computer Analysis of Images and Patterns, pages 26–37. Springer, 2015.

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In The elements of statistical learning, pages 485–585. Springer, 2009.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[19] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In Computer

Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3304–3311. IEEE, 2010.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[21] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In 2013 12th International Conference on Document Analysis and Recognition, pages 560–564, Aug 2013.

[22] Ahmed Lawgali, Maia Angelova, and Ahmed Bouridane. Hacdb: Handwritten arabic characters database for automatic character recognition. In 2013 4th European Workshop on Visual Information Processing (EUVIP), pages 255–259. IEEE, 2013.

[23] Q. V. Le. Building high-level features using large scale unsupervised learning. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 8595–8598, May 2013.

[24] Yann LeCun. The MNIST database of handwritten digits. http://yann.lecun. com/exdb/mnist/, 1998.

[25] Qing Li, Qiang Peng, and Chuan Yan. Multiple vlad encoding of cnns for image classification. Computing in Science & Engineering, 20(2):52–63, 2018.

[26] G. Louloudis, B. Gatos, N. Stamatopoulos, and A. Papandreou. ICDAR 2013 Competition on Writer Identification. In 2013 12th International Conference on Document Analysis and Recognition, pages 1397–1401, Aug 2013.

[27] G. Louloudis, N. Stamatopoulos, and B. Gatos. Icdar 2011 writer identification contest. In 2011 International Conference on Document Analysis and Recognition, pages 1475–1479, Sept 2011.

[28] Georgios Louloudis, Basilis Gatos, and Nikolaos Stamatopoulos. Icfhr 2012 competition on writer identification challenge 1: Latin/greek documents. In Frontiers in handwriting recognition (ICFHR), 2012 international conference on, pages 829–834. IEEE, 2012.

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91–110, 2004.

[30] J Mantas. An overview of character recognition methodologies. Pattern recognition, 19(6):425–430, 1986.

[31] V Märgner, U Pal, A Antonacopoulos, et al. Writer Identification. Series in Machine Perception and Artificial Intelligence, 82:121–154, 2018.

[32] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In International Conference on Artificial Neural Networks, pages 52–59. Springer, 2011.

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In NIPS workshop on deep learning and unsupervised feature learning, volume 2011, page 5, 2011.

[34] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3384–3391, June 2010.

[35] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):63–84, Jan 2000.

[36] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos. ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016). In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 619–623, Oct 2016.

[37] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 1395–1403, Nov 2017.

[38] I. Pratikakis, K. Zagoris, B. Gatos, J. Puigcerver, A. H. Toselli, and E. Vidal. ICFHR2016 Handwritten Keyword Spotting Competition (H-KWS 2016). In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 613–618, Oct 2016.

[39] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold. Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 471–476, Oct 2016.

[40] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal. ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset. In 2016 15th

International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 630–635, Oct 2016.

[41] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara. The state of the art in online handwriting recognition. IEEE Transactions on pattern analysis and machine intelligence, 12(8):787–808, 1990.

[42] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM.

[43] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec):3371–3408, 2010.

[44] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In European Conference on Computer Vision, pages 835–851. Springer, 2016.

[45] Yu-Jie Xiong, Ying Wen, Patrick SP Wang, and Yue Lu. Text-independent writer identification using SIFT descriptor and contour-directional feature. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 91–95. IEEE, 2015.

[46] C. Yang, X. C. Yin, H. Yu, D. Karatzas, and Y. Cao. Icdar2017 robust reading challenge on text extraction from biomedical literature figures (detext). In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 1444–1447, Nov 2017.