






Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Universitat Autònoma  
de Barcelona**

**Document Image Enhancement and Recognition in Low  
Resource Scenarios: Application to Ciphers and  
Handwritten Text**

A dissertation submitted by **Mohamed Ali  
Souibgui** at Universitat Autònoma de Barcelona  
to fulfil the degree of **Doctor of Philosophy**.

Bellaterra, September 21, 2022

Directors

**Dra. Alicia Fornés**

Universitat Autònoma de Barcelona  
Dept. Ciències de la Computació  
Centre de Visió per Computador

**Dr. Yousri Kessentini**

University of Sfax  
SM@RTS Laboratory  
Digital Research Center of Sfax

Thesis  
committee

**Prof. Andreas Maier**

Friedrich-Alexander Universität Erlangen-Nürnberg  
Department of Computer Science  
Pattern Recognition Lab

**Dr. Ernest Valveny Llobet**

Universitat Autònoma de Barcelona  
Dept. Ciències de la Computació  
Centre de Visió per Computador

**Dra. Naila Murray**

Meta AI Research



---

This document was typeset by the author using  $\text{\LaTeX}$ 2 $\epsilon$ .

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

This work is licensed under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) 2020 by **Mohamed Ali Souibgui**. You are free to copy and redistribute the material in any medium or format as long as you attribute its author. If you alter, transform or build upon this work, you may distribute the resulting work only under the same, similar or compatible license.

ISBN 978-84-121011-8-8

Printed by Ediciones Gráficas Rey, S.L.

To the memory of my father,  
my mother,  
my brother,  
and my loved ones ...



# Acknowledgments

Without a collaborative and supportive environment, in which free exchange of ideas and camaraderie emerges, research or any creative endeavor is not possible. Therefore, I would like to acknowledge and express my gratitude to all the people that I have come across, who in one way or another led me to finish this research work.

Firstly, I would like to thank my supervisors Dra. Alicia Fornés and Dr. Youstri Kessentini for guiding my research and giving me support during this thesis long road. Thank you for listening to my ideas, and providing perspectives and criticism at every meeting we had during the last 3 years.

My thanks go also to the Computer Vision Center (CVC) and the DECRYPT project for providing me with a 4 years doctoral study scholarship, without that it would have not been possible. I thank all the professors and doctors that I collaborate with during my thesis, Prof. Josep Lladós, Dr. Dimosthenis Karatzas, Dr. Lluís Gomez, Prof. Beáta Megyesi, Dra. Michelle Waldispühl, and Dra. Sana Khamekhem Jemni.

Andrés, Sanket (A.k.a. Nicho), Sergi, Ali, and Sounak. I was lucky enough to share several years with you in CVC. I met you at the beginning of my Ph.D studentship, and we shared a lot of interesting moments, trips, and discussions and I learned a lot from you. From Andrés, I learned that life is not a competition, to slow down and care more about the people I love. I learned to listen more and work harder from Sergi. Thank you, Sanket for being always there whenever I needed your support, you are a true brother to me. Thank you Ali for always providing us with a lot of wise advice, and drawing our attention to a lot of interesting topics (in AI and in Life).

Also, I would like to thank all my friends from CVC, Ruben, Khanh, Ayan, Laura, Hector, Lei, Giussepe, Francesco, Marco, Emanuelle, David, Manu, Pietro, Enrico, Andrea, Kai, Pau, Pau, Armin, Asma, and Jialuo. I keep all of you in my memories.

Thanks to my co-members within the Decrypt project. We did a lot of interesting collaborations and shared many interesting discussions and ideas. I also would like to thank all the Centre de Visió per Computador (CVC) staff, specially Montse, Gigi, Kevin, and Marc. You provide a valuable service to foreign students and make everything easier.

Special thanks to the people that are close to my heart. To Gofran, thanks for all your love, and friendship that you provided me with during the several years that I stayed in Monastir, your best remains within me. Thanks to my childhood and university friends, Karim, Rania, Lobna, Hosni, Sarah, Ayoub, Hamda, Riadh, Sami and Imen. I owe you friendship and love, forever.

My final thanks go to my family. My brother Mourad thank you for your support. My mother Mabrouka, thank you for your infinite love, and for pushing toward sending me to school, without you I was not going to be educated or accomplish any of my research work. To the memory of my father Monji, you were the most hard-working person I have ever seen. You are not here when I am reaching the end of this thesis, but you were here when I started, without your support and effort this was never going to happen, rest in peace. To the rest of my family and friends, thank you all.

# Abstract

In this thesis, we propose different contributions with the goal of enhancing and recognizing historical handwritten document images, especially the ones with rare scripts, such as cipher documents.

In the first part, some effective end-to-end models for Document Image Enhancement (DIE) using deep learning models were presented. First, Generative Adversarial Networks (cGAN) for different tasks (document clean-up, binarization, deblurring, and watermark removal) were explored. Next, we further improve the results by recovering the degraded document images into a *clean* and *readable* form by integrating a text recognizer into the cGAN model to promote the generated document image to be more readable. Afterwards, we present a new encoder-decoder architecture based on vision transformers to enhance both machine-printed and handwritten document images, in an end-to-end fashion.

The second part of the thesis addresses Handwritten Text Recognition (HTR) in low resource scenarios, i.e. when only few labeled training data is available. We propose novel methods for recognizing ciphers with rare scripts. First, a *few-shot* object detection based method was proposed. Then, we incorporate a progressive learning strategy that automatically assigns pseudo-labels to a set of unlabeled data to reduce the human labor of annotating few pages while maintaining the good performance of the model. Secondly, a data generation technique based on Bayesian Program Learning (BPL) is proposed to overcome the lack of data in such rare scripts. Thirdly, we propose a Text-Degradation Invariant Auto Encoder (Text-DIAE). This latter self-supervised model is designed to tackle two tasks, text recognition and document image enhancement. The proposed model does not exhibit limitations of previous state-of-the-art methods based on contrastive losses, while at the same time, it requires *substantially* fewer data samples to converge.

In the third part of the thesis we analyze, from the user perspective, the usage of HTR systems in low resource scenarios. This contrasts with the usual research on HTR, which often focuses on technical aspects only and rarely devotes efforts on implementing software tools for scholars in Humanities.

**Keywords** – Computer Vision, Historical Document Analysis, Document Image enhancement, Handwritten Text Recognition, Few-shot learning, Generative Adversarial Networks, Transformers.





# Resum

En aquesta tesi proposem diferents contribucions per tal de millorar i reconèixer imatges de documents manuscrits històrics, especialment aquells amb escriptures rares, com els documents xifrats.

A la primera part es presenten alguns models efectius d'extrem a extrem per millorar imatges de documents utilitzant models d'aprenentatge profund. En primer lloc, s'exploren xarxes adversàries generatives (cGAN) per a diferents tasques (neteja de documents, binarització, desenfocament i eliminació de marques d'aigua). A continuació, millorem els resultats recuperant les imatges de documents degradats en un format *llegible* mitjançant la integració d'un reconeixedor de text al model cGAN. Posteriorment, presentem una nova arquitectura de codificador-decodificador basada en *transformers* per millorar les imatges de documents impresos i manuscrits, de manera integral.

La segona part de la tesi aborda el reconeixement de text manuscrit (HTR) en escenaris de baixos recursos, és a dir, quan només hi ha disponibles poques dades etiquetades d'entrenament. Proposem mètodes nous per reconèixer documents xifrats amb alfabetos rars. En primer lloc, es proposa un mètode basat en mètodes de poques dades (*few-shot*) per detectar objectes. Després, incorporem una estratègia d'aprenentatge progressiu que assigna automàticament pseudoetiquetes a un conjunt de dades sense etiquetar per reduir el treball humà d'anotar algunes pàgines mentre es manté el bon rendiment del model. En segon lloc, es proposa una tècnica de generació de dades basada en l'aprenentatge de programes bayesians (BPL) per superar la manca de dades en alfabetos rars. En tercer lloc, proposem un *autoencoder* invariable a la degradació de text. Aquest darrer model autosupervisat està dissenyat per abordar dues tasques, el reconeixement de text i la millora de la imatge del document. El model proposat no presenta les limitacions dels mètodes anteriors basats en *contrastive losses*, mentre que ahora requereix *substancialment* menys mostres de dades per convergir.

A la tercera part de la tesi analitzem, des de la perspectiva de l'usuari, l'ús de sistemes HTR a escenaris de baixos recursos. Això contrasta amb la investigació habitual sobre HTR, que sovint se centra només en aspectes tècnics i poques vegades dedica esforços a implementar eines de programari per a acadèmics en Humanitats.

**Keywords** – Visió per Computador, Anàlisi de documents històrics, millora d'imatges, reconeixement de text manuscrit, aprenentatge amb pocs exemples, xarxes adverses generatives, *transformers*.



# Resumen

En esta tesis proponemos diferentes contribuciones con el objetivo de mejorar y reconocer imágenes de documentos manuscritos históricos, especialmente aquellos con escrituras raras, como los documentos cifrados.

En la primera parte, se presentan algunos modelos efectivos de extremo a extremo para la mejora de imágenes de documentos utilizando modelos de aprendizaje profundo. En primer lugar, se exploran las redes adversarias generativas (cGAN) para diferentes tareas (limpieza de documentos, binarización, desenfoque y eliminación de marcas de agua). A continuación, mejoramos los resultados recuperando las imágenes de documentos degradados en un formato *legible* mediante la integración de un reconocedor de texto en el modelo cGAN. Posteriormente, presentamos una nueva arquitectura de codificador-decodificador basada en *transformers* para mejorar las imágenes de documentos impresos y escritos a mano, de manera integral.

La segunda parte de la tesis aborda el reconocimiento de texto escrito a mano (HTR) en escenarios de bajos recursos, es decir, cuando solo hay disponibles pocos datos etiquetados de entrenamiento. Proponemos métodos novedosos para reconocer cifrados con alfabetos raros. En primer lugar, se propone un método basado en métodos de pocos datos (*few-shot*) para detección de objetos. Luego, incorporamos una estrategia de aprendizaje progresivo que asigna automáticamente pseudoetiquetas a un conjunto de datos sin etiquetar para reducir el trabajo humano de anotar algunas páginas mientras se mantiene el buen rendimiento del modelo. En segundo lugar, se propone una técnica de generación de datos basada en el aprendizaje de programas bayesianos (BPL) para superar la falta de datos en alfabetos raros. En tercer lugar, proponemos un *autoencoder* invariable a la degradación de texto. Este último modelo autosupervisado está diseñado para abordar dos tareas, el reconocimiento de texto y la mejora de la imagen del documento. El modelo propuesto no presenta limitaciones de los métodos anteriores basados en *contrastive losses*, mientras que al mismo tiempo requiere *sustancialmente* menos muestras de datos para converger.

En la tercera parte de la tesis analizamos, desde la perspectiva del usuario, el uso de sistemas HTR en escenarios de bajos recursos. Esto contrasta con la investigación habitual sobre HTR, que a menudo se centra solo en aspectos técnicos y rara vez dedica esfuerzos a implementar herramientas de software para académicos en Humanidades.

**Keywords** – Visión por Computador, Análisis de documentos históricos, mejora de imágenes, reconocimiento de texto manuscrito, aprendizaje con pocos ejemplos, redes adversas generativas, *transformers*.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Historical Documents Analysis . . . . .	1
1.2	Degraded Manuscripts . . . . .	2
1.3	Ciphered Manuscripts . . . . .	3
1.4	Deep Learning in Computer Vision . . . . .	4
1.5	Learning in Low Resource . . . . .	5
1.6	Scope and Research Questions . . . . .	6
1.7	Thesis Structure/ Outline . . . . .	8
<b>I</b>	<b>Document Image Enhancement</b>	<b>11</b>
<b>2</b>	<b>Document Image Enhancement: State-Of-The-Art</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Related Work . . . . .	14
2.2.1	Degraded document enhancement . . . . .	14
	Classic Approaches . . . . .	14
	Energy Based Approaches . . . . .	15
	Deep Learning Approaches . . . . .	15
2.3	Conclusion . . . . .	16
<b>3</b>	<b>Document Image Enhancement Using A Conditional Generative Adversarial Network</b>	<b>19</b>
3.1	Introduction . . . . .	19
3.2	Related Work . . . . .	21
3.2.1	Degraded document Image Enhancement . . . . .	21
3.2.2	Watermark removal . . . . .	21
3.2.3	Generative adversarial networks for image-to-image translation . . . . .	22
3.3	Proposed approach . . . . .	22
3.3.1	Generator: . . . . .	23
3.3.2	Discriminator . . . . .	24
3.3.3	Training process . . . . .	25
3.4	Experiments and results . . . . .	26
3.4.1	Document cleaning and binarization . . . . .	26

3.4.2	Watermark removal . . . . .	29
3.4.3	Comparison with other GAN models . . . . .	32
3.4.4	Document deblurring . . . . .	33
3.4.5	OCR evaluation . . . . .	34
3.5	Conclusion . . . . .	36
<b>4</b>	<b>A Multi-Task Adversarial Network for Handwritten Document Image Enhancement</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Proposed Method . . . . .	40
4.2.1	Generator . . . . .	41
4.2.2	Discriminator . . . . .	41
4.2.3	Handwritten Text Line Recognizer . . . . .	42
4.2.4	Training process . . . . .	43
4.3	Experiments and Results . . . . .	44
4.3.1	Metrics . . . . .	44
4.3.2	Handwritten text databases . . . . .	44
Degraded-KHATT . . . . .	44	
Degraded-IAM . . . . .	45	
4.3.3	Results . . . . .	46
Arabic handwritten texts images recovery . . . . .	46	
Latin handwritten texts images recovery . . . . .	49	
H-DIBCO Competitions . . . . .	53	
Dataset selection for the fine-tuning stage . . . . .	58	
4.4	Conclusion . . . . .	59
<b>5</b>	<b>An End-to-End Document Image Enhancement Transformer</b>	<b>61</b>
5.1	Introduction . . . . .	61
5.2	Related Work . . . . .	63
5.2.1	Document Image Enhancement . . . . .	63
5.2.2	Transformers in Vision and Image Enhancement Tasks . . . . .	63
5.3	Method . . . . .	63
5.3.1	Encoder . . . . .	64
5.3.2	Decoder . . . . .	64
5.3.3	Model Variants . . . . .	65
5.4	Experimental Validation . . . . .	65
5.4.1	Choosing the Best Model Configuration . . . . .	65
5.4.2	Quantitative Evaluation . . . . .	66
5.4.3	Qualitative Evaluation . . . . .	68
5.4.4	Self-attention Mechanism . . . . .	72
5.5	Conclusion . . . . .	73

<b>II Document Image Recognition in Low Resource Data</b>	<b>75</b>
<b>6 Handwritten Text Recognition in Low Resource Data: State-Of-The-Art</b>	<b>77</b>
6.1 Introduction . . . . .	77
6.2 Related Work . . . . .	78
6.2.1 Low Resource Manuscript Recognition: The case of ciphered text . . . . .	78
6.2.2 Handwritten Text Pseudo-Labeling . . . . .	79
6.2.3 Data Augmentation and Generation for Handwritten Text . . . . .	80
6.2.4 Self-Supervised Learning . . . . .	80
<b>7 A Progressive Few Shot Learning Approach for Low Resource Handwritten Text Recognition</b>	<b>83</b>
7.1 Introduction . . . . .	84
7.2 Proposed Approach . . . . .	86
7.2.1 Few-shot Manuscript Matching . . . . .	86
7.2.2 Similarity Matrix Decoding . . . . .	87
7.2.3 Progressive Pseudo-Labeling . . . . .	87
Synthetic Data Generation . . . . .	88
Pseudo-Labeling Process . . . . .	89
7.3 Experiments . . . . .	90
7.3.1 Datasets . . . . .	90
7.3.2 Experimental Setup and Metrics . . . . .	91
7.4 Results . . . . .	91
7.4.1 Annotation Time Consumption . . . . .	92
7.4.2 Pseudo-labeling Performance Analysis . . . . .	93
7.4.3 Selecting Threshold for Pseudo-Labeling . . . . .	94
7.4.4 Is semi-supervised learning worth to use? . . . . .	94
7.5 Conclusion . . . . .	95
<b>8 A One-shot Learning Approach for Compositional Data Generation: Application to Low Resource Handwritten Text Recognition</b>	<b>97</b>
8.1 Introduction . . . . .	97
8.2 Bayesian Program Learning (BPL) . . . . .	99
8.3 Handwritten Symbol Generation with BPL . . . . .	101
8.3.1 Dataset . . . . .	101
8.3.2 Data generation results - Symbol Level . . . . .	101
Qualitative Results . . . . .	101
Human Evaluation . . . . .	103
8.4 Impact of Data Generation for HTR . . . . .	104
8.4.1 Data Generation results - Line Level . . . . .	104
8.4.2 HTR models and Evaluation Metric . . . . .	105
8.4.3 HTR Results . . . . .	107
8.4.4 Latin Handwritten Text . . . . .	109
8.5 Conclusion . . . . .	109



<b>9</b>	<b>A Self-Supervised Transformer Autoencoder for Text Recognition and Document Enhancement</b>	<b>111</b>
9.1	Introduction . . . . .	111
9.2	Method . . . . .	113
9.2.1	pretraining Module . . . . .	114
9.2.2	Fine-Tuning . . . . .	115
9.2.3	Learning Objectives . . . . .	116
9.3	Experiments . . . . .	116
9.3.1	Text Recognition . . . . .	117
9.3.2	Document Image Enhancement . . . . .	121
9.4	Conclusion . . . . .	123
<b>III</b>	<b>User Evaluation of HTR Systems in Low Resource Data</b>	<b>125</b>
<b>10</b>	<b>Evaluation of HTR systems for the Automatic Transcription of Rare Manuscripts from a User perspective: Application to <i>Codex Runicus</i></b>	<b>127</b>
10.1	Introduction . . . . .	128
10.2	Related work . . . . .	129
10.2.1	Transcription methods for historical manuscripts . . . . .	129
10.2.2	Transcription tools and user platforms . . . . .	130
10.3	Transcription methods . . . . .	131
10.3.1	Preprocessing: Binarization and Segmentation . . . . .	132
10.3.2	Learning-based method: LSTM-RNNs . . . . .	132
10.3.3	Learning-free method: Unsupervised Clustering . . . . .	133
	Method description. . . . .	133
	Modalities. . . . .	134
10.3.4	Few-Shot Classification method . . . . .	135
	Method description. . . . .	135
	Adaptation for transcription. . . . .	135
10.3.5	Few-shot Detection method . . . . .	136
	Method description. . . . .	136
	Adaptation for transcription. . . . .	137
10.4	Experiments . . . . .	137
10.4.1	Dataset . . . . .	138
10.4.2	Evaluation Metrics . . . . .	139
10.4.3	Results . . . . .	140
10.4.4	Time Needed for Preparing the Training Data . . . . .	142
10.4.5	User Validation . . . . .	142
10.5	Recommendations for tools in Digital Humanities . . . . .	144
10.5.1	Advantages and Disadvantages . . . . .	144
10.5.2	Recommendations . . . . .	145
	Availability of labeled training data. . . . .	145
	Multi-writer manuscripts. . . . .	145
	Cursive Handwriting. . . . .	146

---

Unknown alphabet . . . . .	146
Length of Manuscript . . . . .	146
10.5.3 Summary . . . . .	147
10.6 Conclusion . . . . .	148
<b>11 Conclusion</b>	<b>151</b>
11.1 Summary of the Contributions . . . . .	151
11.2 Discussion . . . . .	153
11.3 Future Work . . . . .	154
11.3.1 Models Robustness in Document Image Enhancement . . . . .	154
11.3.2 Domain Adaptation . . . . .	154
11.3.3 Continual Learning . . . . .	155
<b>List of Contributions</b>	<b>157</b>
<b>Bibliography</b>	<b>161</b>



# List of Tables

3.1	The obtained results of document cleaning using Noisy office database [203] . . . . .	26
3.2	Results of image binarization on DIBCO 2013 Database. . . . .	27
3.3	Results of image binarization on DIBCO 2017 Database, a comparison with DIBCO 2017 competitors approaches. . . . .	29
3.4	Results of image binarization on DIBCO 2017 and DIBCO 2018 Databases, a comparison with DIBCO 2018 competitors approaches. . . . .	29
3.5	Results of watermark removal . . . . .	31
3.6	Results of image binarization for DIBCO 2018 Database . . . . .	33
3.7	The obtained results of document deblurring . . . . .	34
4.1	Image binarization results for the <i>test set</i> (degraded-KHATT database). (A → B): The CRNN is trained on images from domain A and tested on images from domain B. Deg.: Degraded images. Reco.: Recognition performance. . . . .	46
4.2	Impact of the recognizer weight on the final generated image. . . . .	49
4.3	Image binarization results for the <i>test set</i> (degraded-IAM database). (A → B): The CRNN is trained on images from domain A and tested on images from domain B. Deg.: Degraded images. Reco.: Recognition performance. . . . .	50
4.4	Impact of the proposed binarization method (scenario S1) on the recognition performance by a HTR system. . . . .	50
4.5	Comparative results of our proposed method on <i>H-DIBCO 2012</i> Dataset for document binarization. $Avg = (PSNR + FM + Fps + (100 - DRD)) / 4$ . . . . .	54
4.6	Comparative results of our proposed method on <i>H-DIBCO 2016</i> Dataset for document binarization. $Avg = (PSNR + FM + Fps + (100 - DRD)) / 4$ . . . . .	55
4.7	Comparative results of our proposed method on <i>DIBCO 2017</i> Dataset for document binarization. $Avg = (PSNR + FM + Fps + (100 - DRD)) / 4$ . . . . .	55
4.8	Results for all methods on <i>H-DIBCO 2018</i> Dataset for handwritten document binarization. $Avg = (PSNR + FM + Fps + (100 - DRD)) / 4$ . . . . .	56
4.9	Impact of the fine-tuning data selection on the binarization performance on H-DIBCO <b>2016</b> Dataset. . . . .	59
5.1	Details of our model variants . . . . .	65

5.2	Results of varying the model size for the DIBCO 2017 dataset. †: The higher the better. ‡: The lower the better. . . . .	66
5.3	Results of varying the input and patch sizes for the DIBCO 2017 dataset . .	66
5.4	Comparative results of our proposed method on DIBCO 2011 Dataset. Thresh: Thresholding, Tr: Transformers. . . . .	67
5.5	Comparative results of our proposed method on H-DIBCO 2012 Dataset. Thresh: Thresholding, Tr: Transformers. . . . .	67
5.6	Comparative results of our proposed method on DIBCO 2017 Dataset. Thresh: Thresholding, Tr: Transformers. . . . .	68
5.7	Comparative results of our proposed method on DIBCO 2018 Dataset. Thresh: Thresholding, Tr: Transformers. . . . .	68
7.1	Obtained Results on the different datasets. FT: Fine Tuning. Om: Omniglot. SD: Synthetic Data. RLD: Real Labeled Data. PLD: Pseudo Labeled Data. ULD: UnLabeled Data. . . . .	92
7.2	Required time (in minutes) for manually annotating the training lines. . .	93
7.3	The symbol error rate when using different thresholds while pseudo-labeling the data. Thres.: Threshold . . . . .	94
7.4	Comparative results with self-supervised learning approaches in the semi-supervised scenario. . . . .	95
8.1	Obtained results by different methods and settings: Real and synthetic data were tested with various sizes (# of ann. lines). # of generated samples indicates the number of images per each symbol, used to generate the synthetic lines. . . . .	106
8.2	The results on IAM dataset, simulating the low resource handwritten recognition. The numbers are in terms of character error rate (lower is better). .	109
9.1	<b>Representation quality.</b> We evaluate the encoder capability of learning visual representations. This scenario is analogous as the linear probing in self-supervised models. We train a decoder with labelled data on top of a frozen encoder pre-trained on the proposed degradation. The column <i>Seen</i> refers to the number of samples in millions seen during pre-training. Word prediction in terms of Accuracy (Acc) and single edit distance (ED1) in handwritten and text recognition. . . . .	117
9.2	<b>Semi-supervised results.</b> Accuracy obtained by fine-tuning a pre-trained model with varying percentages of the labeled dataset. Under this setting, we back-propagate the gradients through the specific decoder and the pre-trained encoder. . . . .	118
9.3	<b>Ablations of the pre-training objectives.</b> Results in handwritten and scene-text recognition obtained by each pretext task. The performance is measured in terms of Word and Character error rates (WER and CER). . . . .	119
9.4	<b>SOTA results.</b> Quantitative evaluation with state-of-the-art methods on the IAM word level dataset. . . . .	120

9.5	<b>SOTA results.</b> Comparison of the proposed Text-DIAE compared to previous state-of-the-art approaches on the different DIBCO and H-DIBCO Benchmarks . . . . .	121
9.6	<b>SOTA results:</b> Quantitative evaluation with state-of-the-art methods on the deblurring dataset. . . . .	122
9.7	<b>Ablations of the degradations as pre-training objectives.</b> Results in document image binarization on DIBCO 2018 obtained by each pretext task in terms of PSNR. . . . .	122
10.1	Overview of the characteristics of the different methodologies. . . . .	132
10.2	Codex Runicus manuscript pages used in the different experiments scenarios, the numbers are related to the order of the pages in the original manuscript . . . . .	139
10.3	Obtained results by the transcription methods over the different scenarios. The CER and the Missing Characters are shown in %. Unsupervised clustering methods include with and without User Intervention (UI) . . .	140
10.4	Time consumed for data preparation for training (hours:minutes). . . . .	142
10.5	Number of errors and time needed for validation time vs manual transcription. . . . .	143
10.6	Summary of recommendations for choosing a transcription method. The symbols mean the following. ✗means not suitable, = means medium suitability, and ✓means very suitable. . . . .	148



# List of Figures

1.1	An example of the degradation that can occur in historical documents . . .	2
1.2	Examples of handwritten ciphers dated from the 16th to the 18th century. Top: Devil cipher. Middle: Borg Cipher. Bottom: Copiale cipher. . . . .	4
3.1	Examples of the documents used in this study: (a): Degraded documents, (b): A document with dense watermark. . . . .	20
3.2	The generator follows the U-net architecture [157]. Each box corresponds to a feature map. The number of channels is denoted at the bottom of the box. The arrows denote the different operations. . . . .	24
3.3	The Discriminator architecture . . . . .	25
3.4	The proposed DE-GAN . . . . .	25
3.5	Binarization of degraded documents by DE-GAN, the result is satisfac- tory, except in some parts that were highly dense (the red boxes in the row of the predicted image) . . . . .	27
3.6	Qualitative binarization results produced by different methods of a part from the sample (PR5), which is included in DIBCO 2013 dataset . . . . .	28
3.7	Qualitative binarization results produced by different methods of of a part from the sample (HW5), which is included in DIBCO 2013 dataset . . .	28
3.8	Binarization of three historical degraded documents by DE-GAN, the bi- narized version is presented under each original image. Some parts are not well recovered as shown in the red boxes. . . . .	30
3.9	4 Samples from our developed Dataset . . . . .	31
3.10	Watermark removal by DE-GAN . . . . .	32
3.11	Qualitative results for dense watermark removal. Above, is a section from watermarked invoice. Below, it's enhanced version. Some parts of the text in the invoice were blurred due to privacy constraints. Because of different domains, synthetic vs real, we can see that some tiny parts of the watermark were not completely removed (red boxes). . . . .	33
3.12	Qualitative binarization results produced by different models of the sam- ple (9) from H-DIBCO 2018 dataset . . . . .	34
3.13	Qualitative deblurring results of some patches produced by different meth- ods . . . . .	35
3.14	Qualitative results for Tesseract recognition of some text lines . . . . .	35



4.1	Examples of the degradation that can be appeared in handwritten text images. . . . .	38
4.2	Proposed architecture for document binarization. . . . .	40
4.3	Generator's architecture design used in this study. . . . .	41
4.4	Discriminator's architecture used in this study. . . . .	42
4.5	Workflow of the CNN-Bi-GRU recognizer's architecture. . . . .	43
4.6	Examples of distorted line images of the degraded-KHATT database used in this study, images are presented in gray level. . . . .	45
4.7	Examples of distorted line images of the degraded-IAM database used in this study, images are presented in gray level. . . . .	45
4.8	Results of our proposed method for recovering degraded lines images. (a): GT, (b): Distorted, (c): Baseline cGAN, (d): cGAN [175], (e): Ours S1, (f): Ours S2. . . . .	47
4.9	Results of our proposed method for recovering extremely degraded lines images. (a): GT, (b): Distorted, (c): Baseline cGAN, (d): cGAN [175], (e): Ours S1, (f): Ours S2. . . . .	48
4.10	Results of fixing a degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by the CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images (better viewed in color),R (Generated): recognition by CRNN [134] trained on generated images (S2). . . . .	51
4.11	Results of fixing a highly degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by the CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images, R (Generated): recognition by CRNN [134] trained on generated images (S2). . . . .	52
4.12	Results of fixing an extremely degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images, R (Generated): recognition by the CRNN [134] trained on generated images (S2). . . . .	53
4.13	Results of our method in binarization of some samples from the H-DIBCO 2018 dataset. Images in columns are: Left: original image, Middle: GT image, Right: Binarized image using our proposed method. . . . .	56
4.14	Results of the different enhancements on sample 4, from H-DIBCO 2018 Dataset. . . . .	57
4.15	Qualitative results of our proposed method evaluated on an a part taken from a sample from H-DIBCO 2018 Dataset (Pixels restoration). . . . .	58

5.1	Proposed model: The input image is split into patches, which are linearly embedded, and the positional information is added to them (this is not shown in Figure because of space constraint). The resulting sequence of vectors is fed to a standard Transformer encoder to obtain the latent representations. These representations are fed to another Transformer representing the decoder to obtain the decoded vector, which is linearly projected to vectors of pixels representing the output image patches. . . .	64
5.2	Qualitative results of our proposed method in binarization of some samples from the DIBCO and H-DIBCO datasets. Images in columns are: Left: original image, Middle: GT image, Right: Binarized image using our proposed method. . . . .	70
5.3	Qualitative results of the different binarization methods on the sample number 12 from DIBCO 2017 Dataset. . . . .	71
5.4	Attention maps from the $2^{nd}$ head of the last layer of DocEnTr{8} encoder. We display the self-attention for different (random) tokens. . . . .	72
5.5	Attention maps from the $2^{nd}$ head of the last layer of DocEnTr{8} encoder. We display the self-attention for different (random) tokens. (A failure case). . . . .	73
7.1	Our few-shot approach for handwriting recognition. Examples of each symbol in the alphabet are used as supports. Up: Detection of a support symbol in a handwritten line. Down: Construction of the similarity matrix from the predicted bounding boxes and its decoding to obtain the final text. . . . .	85
7.2	An illustration of the attention RPN: the support feature map is average pooled until obtaining a tensor with the shape of $1 \times 1 \times 512$ . The obtained tensor is multiplied over depth with the Query feature map to obtain the attention Q, which is passed to the RPN for region proposing. . . . .	85
7.3	Our pseudo-labeling approach: In the beginning, synthetic lines are generated using the support set. Then, the pseudo-labeling phase starts. At starting, there is no pseudo-labeled data, so only synthetic lines will be used for retraining the model. Then, the model predicts symbols from the real unlabeled lines with the same script. The symbols with highest confidence score, namely pseudo-labels, are labeled and added with their predicted bounding boxes. Next, the model is retrained again using the synthetic lines and the pseudo-labeled symbols from real lines. The process is repeated until the full dataset is annotated. . . . .	87
7.4	An example of pseudo-labeling of a line image. The background is colored in grey, while the predicted label classes at each time are shown in colors. Each symbol class is shown with a different color (best viewed in color). . . . .	89
7.5	Examples of the three manuscripts with low resource annotated data. . . . .	90

8.1	(A) A generative model of handwritten ciphered symbols. (i) From a library of color-coded primitives, new types are generated, (ii) combining these subparts, (iii) to further generating parts, (iv) and then defining simple programs by combining parts with relations. (v) Running these programs new tokens are generated, (vi) which are then rendered as raw images. (B) An image along with their log-probability scores for the five best programs. Parts are distinguished by color, with a colored flat back indicating the beginning of a stroke and a black arrowhead indicating the end. . . . .	99
8.2	Two lines images from the Borg cipher. The image shows that there are frequent touching symbols in this manuscript, even between different lines. . . . .	101
8.3	Generating new exemplars given one ciphered symbol. (A): Conditioning on the same symbol (in-sample) shown on top of the nine-cipher grids. (B): Conditioning on a different example of the same class (out-sample). The nine-character grids were generated by BPL. . . . .	102
8.4	Result of the AMT human study where subjects are asked to match between real and generated images. The consensus seen in the $x$ -axis represents the amount of agreement among subjects. . . . .	103
8.5	Examples of the three sets of lines, created by concatenating the symbols. . . . .	105
8.6	SER of testing with real Borg lines and synthetic BPLL lines, using different mixing settings and conditioning on different numbers of samples for generation. . . . .	107
9.1	<b>Text-Degradation Invariant Auto-Encoder (Text-DIAE)</b> , we employ image reconstruction pretext tasks at pretraining. Masking, blurring and adding noise are employed to learn richer representations. . . . .	112
9.2	<b>pretraining pipeline.</b> Text-DIAE aims to learn degradation invariant representations. These are later used to reconstruct the input image with a specific learning objective for each degradation type. . . . .	114
9.3	<b>Fine-tuning pipeline.</b> We start from a pretrained encoder as initial weights to solve a specific downstream task. Explicit decoders are used for text recognition (left) and document image enhancement (right). . . . .	115
9.4	<b>Qualitative results of pretraining samples.</b> The left refers to handwritten text, while scene-text is depicted on the right. On each scenario, from left to right, the original, masked and reconstructed images are depicted. . . . .	120
9.5	<b>Qualitative results of deblurred samples.</b> The document image on the left refers to the originally captured blurred image, followed by the ground-truth, and the deblurred results from the DocEnTr and our Text-DIAE model towards the right. The correctly predicted OCR output is shown in "Green" font while the inaccurate ones are depicted in "Red" and recognition performance in terms of CER. . . . .	122

10.1	Examples of elements from two different clusters. The scores shown in green mean that the element is correctly clustered, while the ones in red are wrongly clustered because of the high visual similarity. . . . .	134
10.2	Unsupervised clustering with user intervention. Left: The user removes the wrongly clustered symbol in each cluster. Right: The user assigns the corresponding label/transcription to each cluster. . . . .	135
10.3	Few-Shot classification model. . . . .	136
10.4	Few-shot detection method for transcription: The matching model is applied to construct the similarity matrix between the alphabet and the query line. After that, the matrix is decoded to obtain the final recognized text. . . . .	137
10.5	Codex Runicus, Arnamagnæan Collection, University of Copenhagen, AM 28 8vo, fol. 86r ( <a href="https://www.e-pages.dk/ku/579/">https://www.e-pages.dk/ku/579/</a> ) . . . . .	138
10.6	Transcription results from scenario L2 of page 173, line 13. The first row corresponds to the ground truth provided by the expert. Errors are shown in orange color, and skipped characters in blue (better viewed in color). . .	141
10.7	Estimated total time needed for transcribing the <i>Codex Runicus</i> in scenario L2. We assume that the Few-shot detection method needs 2h for data preparation (6 pages), and 5 min/page for validation; the clustering method with user intervention needs 1'4h for data preparation, and 6'5 min/page for validation; the manual transcription is 14'5 min/page. . . .	147



# Chapter 1

## Introduction

*The past resembles the future more than one drop of water resembles another.*  
– Ibn Khaldun

---

*Historical document analysis has been an active field within the pattern recognition and computer vision community. This Chapter briefly overviews this research field. Then, the problems related to document degradation and the rare manuscripts (and alphabets) are introduced. Followed by an overview of the methods that can be used to tackle this problem based on deep learning, especially in low-resource data scenarios. Finally, we summarize the research questions, objectives, and contributions of this work.*

---

### 1.1 Historical Documents Analysis

Since the invention of writing, spatio-temporal communications between different human beings became possible using manuscripts [58]. Nowadays, we can find many historical documents from different nations in libraries. These documents are carrying valuable information about the history, civilizations, ancient societies, cultures, and scientific reports. Digitizing these rare documents and making them accessible to everyone is an important task since they are considered a global human heritage. Indeed, processing the digitized documents (numerical images) to present them in a good and *enhanced* condition is even more appreciated. Also, automatically extracting their information by *recognizing* them can facilitate many automatic applications such as indexation, search and query, recommendation, storing, etc.

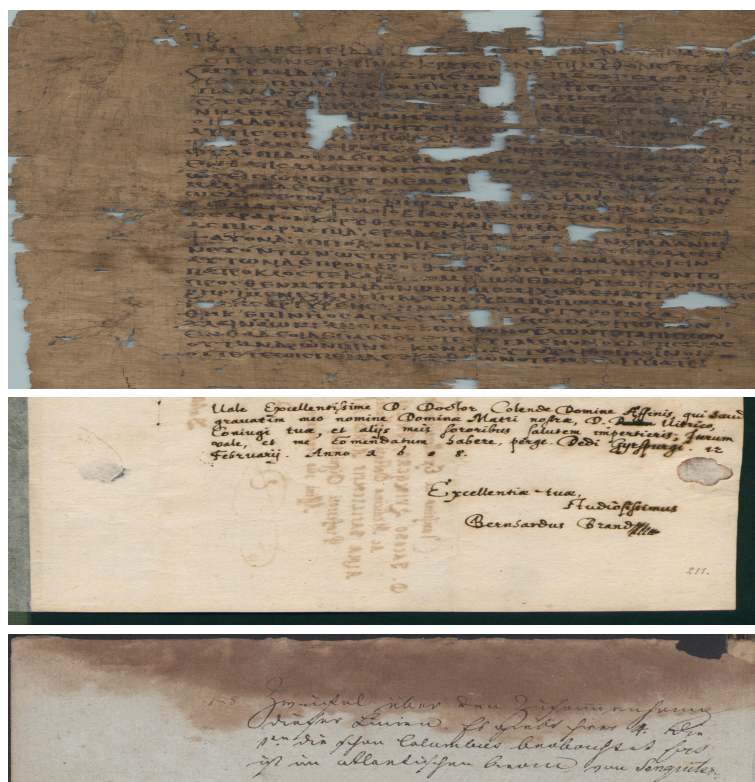


Figure 1.1: An example of the degradation that can occur in historical documents

Research in historical documents is being done within the field called Document Image Analysis and Recognition (DIAR) which is belonging to the Computer Vision and Pattern Recognition (CVPR) research area. Within this field, researchers were addressing many analysis problems, for instance: Handwritten Text Recognition (HTR), Keyword Spotting (KWS), Layout Analysis, Segmentation, Image text alignment, etc. With the recent improvements in machine learning, especially deep learning [107], its architectures have become the central component of most of the proposed approaches for these kinds of document processing problems [121].

## 1.2 Degraded Manuscripts

Historical manuscripts are valuable items. In consequence, access to these documents is only allowed to some expert historians in many cases. This is because of security reasons or also because of the delicate state of some documents when they are too old. Thus, digitizing the documents and presenting them in image format through the in-

Internet is a solution to allow more accessibility. However, many historical document images are coming with *degradation* problems. Degradation can either be occurred because of the document itself (Time effect when the document is too old, wrinkles, stains, ink bleed-through, etc.) or because of a bad scanning condition, like using a phone camera (Blur, shadow, variation of light condition, perspective angle, etc.). Figure 1.1 shows some examples of degraded documents. As can be seen, degradation can obstruct the reading of the document by a human or a machine. Hence, developing a method to enhance the quality of the document before passing it to the reading stage is important.

### 1.3 Ciphred Manuscripts

Ciphred manuscripts form a particular and specific type of handwritten document that contains secret and encrypted messages or instructions. These documents were used in diplomatic, military, scientific, or religious matters. In order to hide their contents, the sender and receiver were usually creating their own method of writing, by transposing or substituting characters, special symbols, or by inventing a completely new alphabet of symbols to replace the regular one. Some examples of the ciphred manuscripts are depicted in Figure 1.2. In the national archives and libraries, 1% of the records contain encrypted manuscripts [128]. Historical Ciphred Manuscripts in the libraries and archives are seldom indexed as ciphers, which makes them hard to be found. However, these documents hold valuable information about our past and serve most of the time as major keys to reinterpreting it.

Given the difficulties in the decryption of such manuscripts, some multi-disciplinary initiatives, such as the Decrypt project<sup>1</sup>[129], have emerged to join the expertise in computer vision, computational linguistics, philology, cryptoanalysis, and history to make advances in historical cryptology. These joint efforts aim to ease the collection, transcription, decryption, and contextualization of historical ciphred manuscripts in order to unlock their contents and make the secret information available for scholars in history, science, religion, etc.

This thesis has been carried out in the framework of the above-mentioned Decrypt project [129], with the goal of enhancing and transcribing historical ciphred document images. The main obstacles that we encountered were data scarcity, both in terms of the number of documents and the available labeled data. Hence, developing models that work in low-resource scenarios (few data) is necessary.

---

<sup>1</sup><https://de-crypt.org/>



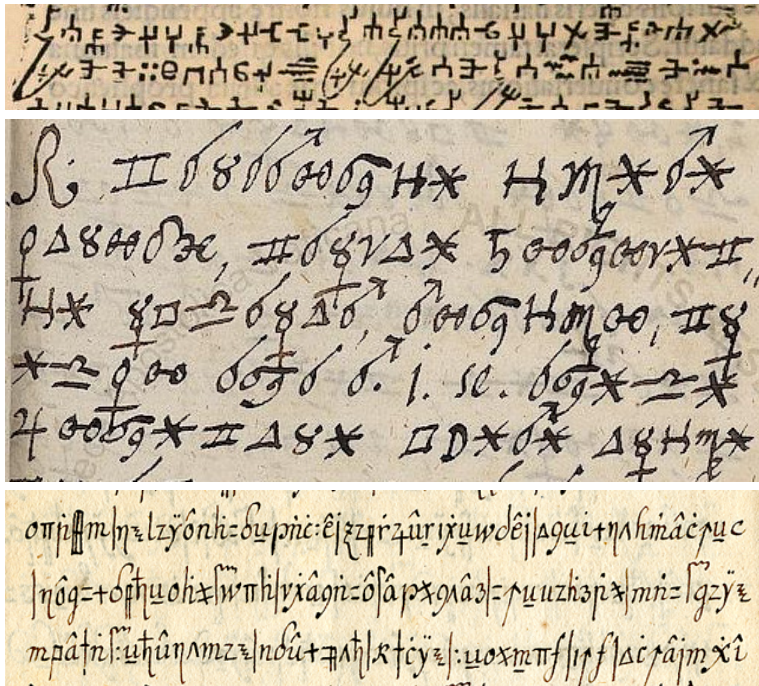


Figure 1.2: Examples of handwritten ciphers dated from the 16th to the 18th century. Top: Devil cipher. Middle: Borg Cipher. Bottom: Copiale cipher.

### 1.4 Deep Learning in Computer Vision

Deep learning approaches started gaining attention in computer vision with the AlexNet in the 2012 and Imagenet Large Scale Visual Recognition Challenges (ILSVRC) [105]. In contrary to the classic handcrafted feature extraction methods, deep learning aims to learn high-level features from raw image pixels in an end-to-end fashion.

Convolutional Neural Networks (CNN) [108] are one of the most used deep network types for image processing tasks. CNN is made up of several cascading layers, with the output of one layer serving as the input for the following one. A stack of linear filters (convolution) is often found in a CNN. Each neuron in a convolutional layer only gets input from a small region of the preceding layer known as the neuron's receptive field. For document image enhancement, convolutional layers were successfully applied in autoencoders [183] and Generative Adversarial Networks (GANs) [174]. Also, CNN was applied in the HTR task along with Recurrent Neural Networks (RNN) [95].

RNN is a type of neural network that deal with sequence data. Due to its internal memory, it is able to recall its input. Thus, The output of a current step is depending explicitly on the output of the previous states. This latter is called the hidden state.

The improvements of the CNN and RNN models over the classic approaches are mainly because of giving to the models the ability to learn rich high-level features from the raw data instead of handcrafting them. However, human priors were still composing a crucial part of those models. This can be seen by forcing the RNN to encode the data as a sequence (one token at each time step) or using the receptive fields in CNN (each filter can only see its local information without directly attending to the other filters in the same convolutional layer). Thus, the transformer [185] model was recently introduced and outperform the RNN [50] in sequence data and CNN [54] in images tasks. The main idea behind transformers is to give more freedom to the model to build its own encoding of the data using its self-attention mechanism (all the sequence of data is encoded in parallel and each token (or patch in images) can attend to the rest of the tokens).

Overall, approaches based on deep CNNs have become favorable in recent years, significantly outperforming traditional approaches in most image processing and recognition tasks. A drawback of deep learning models, as opposed to shallow-learning approaches, is that it requires a large amount of training data. Hence, A part of this thesis is dedicated to finding the mechanisms of training with limited annotated data (low resource).

## 1.5 Learning in Low Resource

In some machine learning problems, data is limited and hard to obtain, especially in an annotated form. Thus, machine learning paradigms had to adapt and learn from one or few data examples per class [56]. In the case of having one labeled example per class, it is called one-shot learning [187]. While in the case of having few labeled examples (usually up to five) it is called Few-Shot Learning (FSL) [192]. In this learning scenario, prior knowledge is used to address the lack of data issue. FSL methods can be categorized into the following three types [192] depending on how to solve the problem:

- **Data:** In this category, prior knowledge is used to perform data augmentation [213, 171], pseudo-labeling (use strategies to automatically annotate a set of unlabeled data belonging to the same domain [173]), or gathering some labeled samples with the same classes and similar domain [184]. Then, the newly acquired data using these techniques are added to enrich the original training set and the full data is fed to a machine learning model.
- **Model:** Here, we use prior knowledge to reduce the size of the hypothesis space. This can be done by different strategies. For instance:
  - Multi-task Learning: when having a set of tasks to learn where in each we have only a few samples per class, we can design a network that used the full data from all the tasks. The model employs some shared weight between

the tasks to capture the generic information while having some layers for task-specific learning. In this way, we benefit from the global information to reduce the hypothesis space for all the tasks [209].

- **Embedding Learning:** In this strategy, the models are designed to embed the samples in a lower-dimensional space. Then, a smaller hypothesis space is constructed by simplifying the learning task, for example by transforming the classification problem into a matching problem by the matching networks [187] or the prototypical networks [169].
- **Algorithm:** here, we use prior knowledge to find a better start for the search hypothesis. For example, by fine-tuning (or refining the parameters) of a pretrained model in a supervised way on a similar task (or data) [133]. Or by using Self-Supervised Learning (SSL) pertaining to learn useful information from a large set of unlabeled data from the same domain, and then perform the fine-tuning on the few data samples [2].

In this thesis, few-shot learning and self-supervised learning mechanisms have been largely explored to deal with the lack of data in historical ciphers and handwritten documents.

## 1.6 Scope and Research Questions

Processing historical manuscript images (quality enhancement and text recognition) is still a challenging problem, in addition, ciphered manuscripts are adding more difficulties. This gives rise to several core research questions addressed by this thesis:

**Research Question 1:** Can we develop and efficiently implement deep neural network based models for image quality enhancement to serve as a preprocessing step for them to better recognize their handwritten text?

**Objective:** An end-to-end framework for image quality enhancement that can recover highly degraded images by removing the degradation while maintaining their readability.

**Contribution:** To address this question by the following contributions:

- An end-to-end document image enhancement model based on conditional GANs. We designed a simple yet effective architecture that can be used to restore different kinds of degradation. We tested our model for document cleaning (binarization) and watermark removal. We showed also that the results of an OCR applied to the enhanced images by our model is improved compared to inputting the original degraded images.
- A major drawback of the document image enhancement models is to delete important parts of the text while cleaning the image. To address this, we designed

a new training objective that takes into account the readability of the document rather than only the visual part. This was done by adding a recognizer in the previous cGAN architecture. Moreover, we did a study to find the best trade-off between the visual and textual features. This results in producing images that are as clean and readable as possible.

- Same as the recent related work in document image enhancement, the proposed models in the two previous chapters were based on the convolutional layers. However, the current state-of-the-art in image processing tasks is using transformers. Thus, we designed and implemented a new model based on a pure transformer architecture. The results obtained by using this model demonstrate its superiority over the previous approaches. The reason behind this superiority is the self-attention mechanism that is employed in the transformer layers and gives the ability to enhance the local patches within the image while having a global view over the rest of the patches.

**Research Question 2:** Can we develop and efficiently implement deep neural network based models for historical manuscript image recognition in low resource scenarios (for example, for ciphered manuscripts)?

**Objective:** To design and implement models that can work on the very few datasets that we have in the case of ciphered and historical manuscripts.

**Contribution:** To address this question by the following contributions:

- A few-shot learning-based model that can recognize historical manuscript images with an unseen alphabet during training. The model requires only one of a few annotated characters/symbols of the new alphabet belonging to the inputted data.
- A progressive and automatic pseudo-labeling approach that can benefit from the unlabeled data. We designed a learning strategy that labels at each iteration a set of new symbols within the unlabeled line to end up in the end by fully pseudo-labeled lines. We showed that this strategy results in improving the model performance by reducing the human effort of annotating data.
- A data generation approach based on the Bayesian Program Learning (BPL) [106] that can generate new training data using only one symbol/character image per each class of the desired manuscript alphabet that will be recognized. The model is merged with the previous FSL model to obtain better results.
- A SSL approach based on the pretraining/fine-tuning fashion. Well-designed learning objectives were used for the pretraining phase to learn rich and useful representations from the unlabeled data. After that, a fine-tuning stage is used and we demonstrate that we get better results than starting from scratch, especially when reducing the amount of training data.

**Research Question 3:** From a user perspective, is it worth using the developed model for the automatic recognition of historical manuscripts?

**Objective:** Demonstrate the utility of our developed models in recognizing the historical text image than transcribing the documents manually.

**Contribution:** We developed and tested four different automatic transcription methods that can be used to train and test models with a limited amount of data. After that, we have exhaustively evaluated these methods both quantitatively and qualitatively with regard to their performance. We have discussed the advantages and disadvantages of each method as well as their applicability from the user perspective considering time consumption in comparison with a fully manual transcription and opportunities for user validation. On the basis of this evaluation, we give recommendations for further development and usage of transcription tools for this kind of manuscript.

## 1.7 Thesis Structure/ Outline

This thesis structure is composed of an introduction, and three major parts, each containing several related chapters. Then, a conclusion is given with a discussion and future work. In the following, we present a brief description of each part and chapter:

### Chapter 1 – Introduction

In this Chapter we present an introduction of the thesis, we define the problems that we are addressing within the field of historical manuscript images and the faced challenges. We also formalized the research questions that we aim to answer within the thesis along with our contributions.

### Part I – Document Image Enhancement

In this part, we address the document image enhancement problem using deep learning tools. This part is composed of the following four chapters.

- **Chapter 2 – Document Image Enhancement: State-Of-The-Art:** This chapter detail the state-of-the-art in the field of document image enhancement.
- **Chapter 3 – Document Image Enhancement Using A Conditional Generative Adversarial Network:** This chapter proposes an approach for document image enhancement (cleaning, binarization, and watermark removal) based on the conditional generative adversarial networks.
- **Chapter 4 – A Multi-Task Adversarial Network for Handwritten Document Image Enhancement:** In this chapter, we propose an approach for handwritten document image enhancement with an improved readability enhancement compared to the approach proposed in the previous chapter.
- **Chapter 5 – An End-to-End Document Image Enhancement Transformer:** In

this chapter we propose our approach for handwritten document image enhancement using the transformers architecture.

### **Part II – Document Image Recognition in Low Resource Data**

This part addresses handwritten text recognition in a low-resource scenario. It is composed of the following chapters.

- **Chapter 6 – Handwritten Text Recognition in Low Resource Data: State-Of-The-Art:** This chapter detail the related work in handwritten text recognition when facing a low resource scenario.
- **Chapter 7 – A Progressive Few Shot Learning Approach for Low Resource Handwritten Text Recognition:** In this chapter, we introduce the few-shot learning model for text recognition as well as the progressive pseudo-labeling strategy to benefit from the unlabeled data.
- **Chapter 8 – A One-shot Learning Approach for Compositional Data Generation: Application to Low Resource Handwritten Text Recognition:** This chapter introduces the data generation technique for low resource scripts. We detail the method and used the generated lines to train different models performing the HTR task.
- **Chapter 9 – A Self-Supervised Transformer Autoencoder for Text Recognition and Document Enhancement:** This chapter presents the self-supervised and the used pretext tasks to learn useful representation from unlabeled data.

**Part III – User Evaluation of HTR Systems in Low Resource Data:** This part is dedicated to the evaluation of HTR methods from a user perspective.

- **Chapter 10 – Evaluation of HTR systems for the Automatic Transcription of Rare Manuscripts from a User perspective: Application to *Codex Runicus*:** In this chapter, we evaluate the HTR methods applied to a low resource scenario. We used the rare manuscript called Codex Runuius for this study.

**Chapter 11 – Conclusion:** In this chapter, we present the conclusion by summarizing our contributions during this thesis, discussing their limitations, and proposing future work.



## **Part I**

# **Document Image Enhancement**





# Chapter 2

## Document Image Enhancement: State-Of-The-Art

---

*Documents often exhibit various forms of degradation, which make it hard to be read and substantially deteriorate the performance of an OCR system. In this Chapter, we detail the related work to document image enhancement.*

---

### 2.1 Introduction

The preservation and legibility of document images (especially the historical ones) are of utmost priority for the Document Image Analysis and Recognition (DIAR) research. Document records usually contain significant information and in the historical cases it dates back centuries and decades [128]. The conservation of document records can be hampered by several kinds of degradation such as smears, stains, artefacts, pen strokes, bleed-through effects and uneven illumination. These distortions could heavily impact the subsequent downstream tasks for information processing, such as segmentation, Optical Character Recognition (OCR), information spotting and layout analysis. This manifests the need for a robust pre-processing task that denoises and reconstructs a high-quality clean image from its already degraded counterpart. Document Image Enhancement (DIE) aims towards restoring the quality of the degraded document samples to yield a clear enhanced version that is locally uniform.

Automatic document processing consists in the transformation into a form that is comprehensible by a computer vision system or by a human. Thanks to the development of several public databases, document processing has made a great progress

in recent years [33, 165]. However, this processing is not always effective when documents are degraded. Lot of damages could be done to a document paper. For example: Wrinkles, dust, coffee or food stains, faded sun spots and lot of real-life scenarios [203]. Degradation could be presented also in the scanned documents because of the bad conditions of digitization like using the smart-phones cameras (shadow [12], blur [36], variation of light conditions, distortion, etc.). Moreover, some documents could contain watermarks, stamps or annotations. The recovery is even harder when certain types of these later take the text place for instance in cases where the stains color is the same or darker than the document font color. Hence, an approach to recover a clean version of the degraded document is needed.

In this part, we focus on the enhancement of degraded document images by providing models that are used for different recovery tasks: document clean up, binarization, and watermark removal. From a document analyst perspective, this recovery of a clean version from the degraded document falls in the research field called document image enhancement. Where we can find, in addition to those three tasks, other ways to enhance a document image. For instance: unshadowing [12, 101], super-resolution [139], deblurring [36], dewarping [182], etc. In what follows, we cover some related works to our addressed tasks.

## 2.2 Related Work

### 2.2.1 Degraded document enhancement

Degraded document enhancement is related to document image binarization. Where the goal is to produce a binary but clean document. The idea is to classify the pixels of the document as one of two categories: degradation or text. Afterward, assigning zeros to the text pixels and ones for the degradation will generate a binary clean image. While generating a gray-scale or colored image can be done by preserving the same value for the text pixels. Within our work, we aim to recover images that contain hard degradation by removing the background noise, while keeping its readability by OCR and HTR systems as accurate as possible.

#### Classic Approaches

Early image binarization techniques were basing on thresholding. Methods known by global binarization, aimed to find a single threshold value for the whole document. A more sophisticated approaches, named local binarization, determine a different threshold value for each pixel [138, 164, 135, 143, 43, 39]. According to the threshold(s), pixels are classified to be belonging on the text (zero) or the degradation (one). Lelore et al. [114] presented an algorithm called FAIR, based on edge detection to localize the text in a degraded document image. A global threshold selection method was proposed in [5], basing on fuzzy expert systems (FESs), the image contrast is enhanced. Then,

the range of the threshold value is adjusted by another FES and a pixel-counting algorithm. Finally, the threshold value is obtained as the middle value of that range. A machine learning based approach was proposed in [41], the goal was the determination of the binarization threshold in each image region given a three-dimensional feature vector formed by the distribution of the gray level pixel values. The support vector machine (SVM) was used to classify each region into one of four different threshold values. An other and similar SVM based approach was introduced in [198]. The main drawbacks of these classic methods is that the results are highly sensitive to document condition. With a complex image background or a non uniform intensity, problems occurred [146].

### **Energy Based Approaches**

Later, evolved techniques were proposed. Moghaddam et al. [131] proposed a variational model to remove the bleed-through from the degraded double-sided document images. For the cases where the verso side of the document is not available, a modified variational model is also introduced. By transferring the model to the wavelet domain and using the hard wavelet shrinkage, the interference patterns was removed. Other energy based methods were also introduced. In [76], authors considered the ink as a target and tried to detect it by maximizing an energy function. This technique was applied also for scene text binarization [130], which is a similar task. Similarly, Xiong et al. [197] estimated the background and subtracted it from the image by a mathematical morphology. Then the Laplacian energy based segmentation is performed on the enhanced document image to classify the pixels. Although these sophisticated image processing techniques, document binarization results are still unsatisfactory.

### **Deep Learning Approaches**

Recently, deep learning architectures were used to tackle this problem by training their weights directly from raw data. In [3], the problem was formulated as pixels classification depending on sequences. Hence, a 2D Long Short-Term Memory (LSTM) was used to predict each pixel value whether belonging to the text or the degradation given a sequence of its neighbours. This process is, of course, time consuming. Thus, instead of classifying each pixel separately, images were mapped in an end to end fashion from the degraded version into the clean one using the Convolutional Neural Networks (CNNs). These architectures, called auto-encoders, lead to recent improvements in image denoising [126] and more particularly documents binarization [123, 27, 4] or deblurring problems [80]. This kind of applications are now called image-to-image translation, since the goal is to start from a degraded image and learn a mapping function that translate it into a clean domain. Following these approaches, [96] proposed an auto-encoder architecture that performs a cascade of pre-trained U-Net models [157] to learn the binarization with less data. Also, [75] proposed a neural network to learn the the enhancement/binarization in an iterative fashion.

Other deep learning approaches modeled the problem as a generation task, where the goal is to generate a clean version of the image by conditioning on the degraded one. This process was carried out using GANs architectures, composed of a generative model that produce a clean version of the image and a discriminator to assess the binarization result. Thus, and motivated by other approaches where GANs significantly surpasses autoencoders in image-to-image tasks [85], some approaches applying this method were introduced. In [174], a conditional GAN approach was developed for document enhancement and achieved good results in recovering handwritten documents with several backgrounds degradation scenarios, it was also used for optical documents deblurring and dense watermarks removal. A similar cGAN based method was also proposed in [211], where the binarization performed by the generator was done in two stages, learning the pixels in different scales then combining the results to provide the final output. In [46], a strokes preservation method was developed using a GAN model, this was done by learning the text boundary in an auxiliary task for a better document binarization, especially with weak or ambiguous strokes. Another GAN's based method was proposed in [20] using a two networks frameworks, for document binarization. The first one was conditioning on the clean image to generate a degraded one, while the other network reconstructed the clean version conditioning on the degraded image. Thus, an unpaired data training was performed leading to good results when using the second network to binarize images. Similarly, [181] treated the problem as two stages: The first stage was devoted to augment the data by creating degraded handwritten images using a GAN model, while the second stage exploited the generated images to train an inverse problem binarization model.

## 2.3 Conclusion

To summarize, deep learning based methods are now significantly surpassing the classic or modern image processing handcrafted algorithms for the handwritten document binarization task. Thus, we are proposing in Chapter 3 a cGAN based method that achieve good performance in many different degradation recovery tasks.

The only limitation that can be noticed in such deep learning methods is the "*image only*" based training. Because, the usual benchmarks for testing the binarization performance do not include a text recognition evaluation (a GT truth text of the degraded documents). Thus, those methods could easily delete some parts of the handwritten text while binarizing the image, without noticing. It is to note that we addressed this problem in a previous work [175], but for printed documents domain and using the Tesseract OCR engine to evaluate the produced text. Where the character error rate of the OCR on the generated image is inputted as an additional input channel to the discriminator. Moreover, we are proposing in Chapter 4 a handwritten text recognizer that is jointly trained in the GAN architecture to maintain the text, while cleaning the degraded image. Thus, it is more flexible to be used for different handwritten languages and writing styles.

Next, motivating by the success of the recently proposed transformers models [185], we are proposing an other model using Vision Transformers (ViT) [54] to address the document image enhancement problems. This is presented in Chapter 5.



# Chapter 3

## Document Image Enhancement Using A Conditional Generative Adversarial Network

---

*In this Chapter, we propose an effective end-to-end framework named Document Enhancement Generative Adversarial Network (DE-GAN) that uses conditional GANs (cGANs) to restore severely degraded document images. To the best of our knowledge, this practice has not been studied within the context of generative adversarial deep networks. We demonstrate that, in different tasks (document clean-up, binarization, deblurring, and watermark removal), DE-GAN can produce an enhanced version of the degraded document with high quality. In addition, our approach provides consistent improvements compared to state-of-the-art methods over the widely used DIBCO 2013, DIBCO 2017, and H-DIBCO 2018 datasets, proving its ability to restore a degraded document image to its ideal condition. The obtained results on a wide variety of degradation reveal the flexibility of the proposed model to be exploited in other document enhancement problems.*

---

### 3.1 Introduction

In this Chapter, we focus on two document enhancement problems. Degraded documents recovery, i.e., to produce a clean (grayscale or binary) version of the document given any type of degradation, and watermark removal. The faced obstacles are as follows: Overlaps of noise or watermarks with the text, dense watermarks, intense dirt



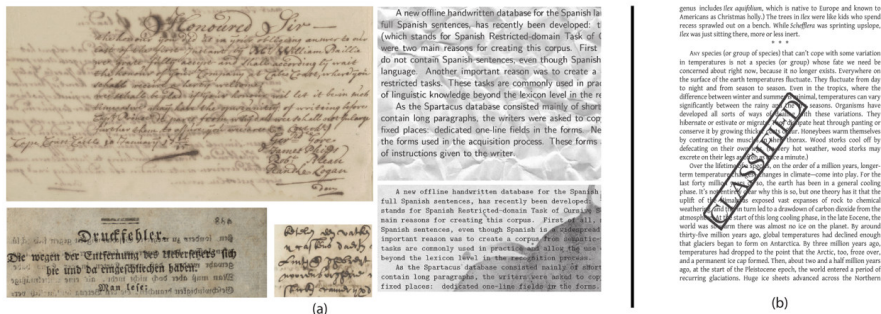


Figure 3.1: Examples of the documents used in this study: (a): Degraded documents, (b): A document with dense watermark.

or degradation can cover the entire text and reading it becomes very hard, there is no prior knowledge about the degradation or the watermark that should be removed, etc. An ideal system should be good in performing two tasks simultaneously, removing the noise and the watermarks as well as retaining the text quality in the document images.

Recently, a great success is made by deep neural networks in natural images generation and restoration, especially deep convolutional neural networks (auto-encoders and variational auto-encoders (VAE)) [126, 52, 99] and generative adversarial networks (GANs) [85, 97]. GANs, which were introduced in [68], are now considered as the ideal solution for image generation problems [97] with high quality, stability and variation compared to the auto-encoders. Generative models gained more attention because of their ability to capture high dimensional probability distributions, imputation of missing data, and deal with multimodal outputs. Despite that, the document analysis research community is not benefiting enough from those approaches, yet. Using them is very limited, for instance, in font translation [19], handwritten profiling [65] and staff-line removal from music score images [104], where promising results were found.

In [85], Isola et al. show that conditional generative adversarial networks (cGANs), a variation of GANs, perform well in image-to-image translation (labels to facade, day to night, edges to photo, BW to color, etc.). While GANs learn a generative model of data, conditional GANs (cGANs) learn a conditional generative model, where it conditions on an input image and generate a corresponding output image. Since document enhancement follows the same process, which means, we want to preserve the text and remove the damage in a conditioned image, cGANs shall be the suitable solution, and this is what motivated us for this study.

The main contributions of this Chapter are: As primary, to the best of our knowledge, this is the first occurrence of GANs, conditional GANs specifically, in a framework that addresses different document enhancement problems (clean up, binarization, and watermark removal). Second, we used a simple but flexible architecture that

could be exploited to tackle any document degradation problem. Third, we introduce a new document enhancement problem consisting in dense watermark and stamp removal. Finally, we experimentally prove that our approach achieves a higher performance compared to the state-of-the-art methods in degraded document binarization.

The rest of the Chapter is organized as follows. Section 3.2, a summary of previous works on document enhancement, especially for document clean-up and binarization and watermark removal in documents as well as in natural images. We review also some related works using the GANs in image-to-image translation. Then, we provide our proposed approach in Section 3.3. Some experimental results and comparisons with traditional and recent methods are described in Section 3.4. Finally, a conclusion with some future research directions is presented in Section 3.5.

## **3.2 Related Work**

### **3.2.1 Degraded document Image Enhancement**

The reader can refer to Section 2.2 for this part.

### **3.2.2 Watermark removal**

Watermark removal is also related to classical document binarization or image matting, where the goal is to decompose a single image into background and foreground knowing that this time the text is in the background while the watermark is in the foreground. But, this problem was not proposed in document processing. In fact, the appeared works that deal with watermark removal were in natural image processing. In [199], the authors used image inpainting algorithms to remove the watermark. Before that, a statistical method was used to detect the watermark region. Dekel et al. [49] proposed to estimate the outline sketch and alpha matte of the same watermark from a batch of different images. Two watermarks were used in this study, the goal was to test the effectiveness of a single visible watermark to protect a large set of images. Wu et al. [194] used the generative adversarial networks [68] to remove watermarks from face images used in a biometric system. Cheng et al. [38] proposed a method based on convolutional neural networks (CNN). First, object detection algorithms were used to detect the watermark region in natural images and then pass it to another model to remove the watermark. In our study, we investigate for the first time the problem of watermark removal in document images, this leads us to compare our approach with some results obtained on natural images for the same purpose.

### 3.2.3 Generative adversarial networks for image-to-image translation

As mentioned above, GANs are now achieving impressive results in image generation and translation. In this paragraph, we investigate the use of this mechanism in related problems to document processing and enhancement. This shall give intuition to the document analysis community about exploiting GANs for these tasks. In [124], it was demonstrated that GANs lead to improvements in semantic segmentation. Ledig et al. presented SRGAN [109], a Generative Adversarial Network for image Super-Resolution. Through it, they achieved photo-realistic reconstructions for large upscaling factors ( $4\times$ ). In [109], conditional GANs were used for several image-to-image translation tasks (these tasks are related to document enhancement), given paired data. This work was extended to [214], where CycleGAN, a GAN that uses impaired data, was proposed as a solution. An other model called "pix2pix-HD" and deals with high-resolution (e.g.  $2048\times 1024$ ) photorealistic image-to-image translation tasks was appeared in [191]. Furthermore, an unsupervised method for image-to-image translation was proposed in [201], where authors train two GANs, or "DualGAN" as they denoted. In their architecture, the primal GAN learns to translate images from a domain  $U$  to a domain  $V$ , while the dual GAN learns to invert the task. The closed loop architecture allows images from each domain to be translated and then reconstructed. Hence, a loss function that accounts for the reconstruction error of images can be used to train the translators.

## 3.3 Proposed approach

We consider the problems of document enhancement as an image-to-image translation task where the goal is to *generate* clean document images given the degraded ones. Since GANs have outperformed auto-encoders in generating high fidelity samples and while we are using paired data, we propose to use a cGAN. We called our model *DE-GAN* (for Document Enhancement conditional Generative Adversarial Network). GANs were initially proposed in [201] and consist of two neural networks, a generator  $G$  and a discriminator  $D$  characterized by the parameters  $\varphi_G$  and  $\varphi_D$ , respectively. The generator has the goal of learning a mapping from a random noise vector  $z$  to an image  $y$ ,  $G_{\varphi_G}: z \rightarrow y$ . While the discriminator has the function of distinguishing between the image generated by  $G$  and the ground truth one. Hence, given  $y$ ,  $D$  should be able to tell if it is *fake* or *real* by outputting a probability value,  $D_{\varphi_D}: y \rightarrow P(\text{real})$ . Those two networks compete against each other in a min-max game, in other words, if one wins the other loses. The generator aims to cheat the discriminator by producing a close image to the ground truth, however, the discriminator will improve his prediction of the image being fake, and this is what is called adversarial learning. cGANs follow the same process, except that, they introduced an additional parameter  $x$ . Which is the conditioned image. Here, the generator is learning the mapping from an observed image  $x$  and a random noise vector  $z$ , to  $y$ ,  $G_{\varphi_G}: \{x, z\} \rightarrow y$  and the discriminator is looking, also, to the conditioned image which makes his process

as  $D_{\varphi_D}: \{x, y\} \rightarrow P(real)$ .

In our situation, the generator will generate a clean image denoted by  $I^C$  given the degraded (or watermarked) one which we will denote  $I^W$ . The generator aims, of course, to produce an image that is very close to the ground truth image denoted by  $I^{GT}$ . The training of cGANs for this task is done by the following adversarial loss function:

$$L_{GAN}(\varphi_G, \varphi_D) = \mathbb{E}_{I^W, I^{GT}} \log[D_{\varphi_D}(I^W, I^{GT})] + \mathbb{E}_{I^W} \log[1 - D_{\varphi_D}(I^W, G_{\varphi_G}(I^W))] \quad (3.1)$$

Using this function, the generator should produce, after several epochs, a similar image to the ground truth, i.e., the watermark and the degradation will be removed and this may fool the discriminator. But, it is not guaranteed that the text will be preserved in a good condition. To overcome this, we employ an additional log loss function between the generated image and the ground truth, for the purpose of forcing the model to generate images that have the same text as the ground truth. It is to note also that this additional loss boosts the training speed, the added function is:

$$L_{log}(\varphi_G) = \mathbb{E}_{I^{GT}, I^W} [-(I^{GT} \log(G_{\varphi_G}(I^W)) + ((1 - I^{GT}) \log(1 - G_{\varphi_G}(I^W)))] \quad (3.2)$$

Thus, the proposed loss of our network, denoted by  $L_{net}$  becomes:

$$L_{net}(\varphi_G, \varphi_D) = \min_{\varphi_G} \max_{\varphi_D} L_{GAN}(\varphi_G, \varphi_D) + \lambda L_{log}(\varphi_G) \quad (3.3)$$

Where  $\lambda$  is a hyper-parameter that was set to 100 for text cleaning and 500 for watermark removal and document binarization. The architecture of generator and discriminator networks is described in the next sections.

### 3.3.1 Generator:

The generator is performing an image-to-image translation task. Usually, auto-encoder models are used for this problem [122, 196, 11]. These models consist, mostly, of a sequence of convolutional layers called an encoder which performs down-sampling until a particular layer. Then, the process is reversed to a sequence of up-sampling and convolutional layers called a decoder. There are two disadvantages of using an encoder-decoder model for the proposed problem: First, due to down-sampling (pooling), a lot of information is lost and the model will have difficulties in recovering them later when predicting an image of the same size as the input. Second, image information flow passes through all the layers, including the bottleneck. Thus, sometimes, a huge amount of unwanted redundant features (inputs and outputs are sharing a lot of the same pixels) are exchanged. Which leads to energy and time loss. For this reason, we employ skip connections following the structure of the model called U-net [157]. Skip connections are added every two layers to recover images with less deterioration, it is to note also that skip connection are used when training a very deep model to prevent

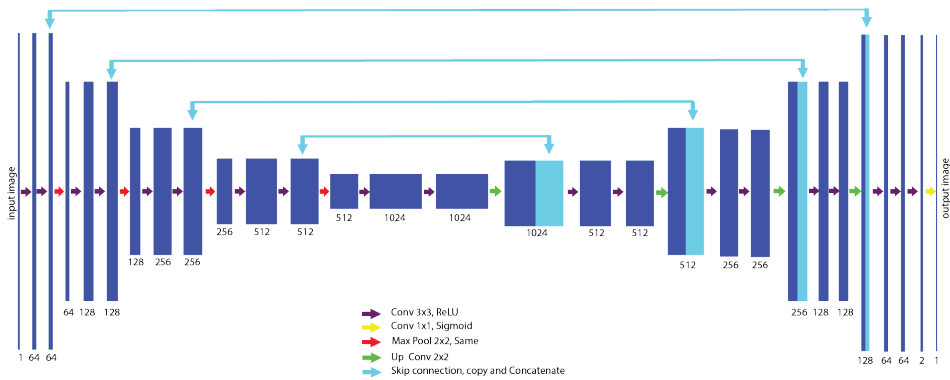


Figure 3.2: The generator follows the U-net architecture [157]. Each box corresponds to a feature map. The number of channels is denoted at the bottom of the box. The arrows denote the different operations.

the gradient vanishing and exploding problems. Some batch normalization layers are also added to accelerate the training. The architecture of the generator used in this study is illustrated in Fig. 3.2, it is similar to [157] where it was introduced for the purpose of biomedical image segmentation.

### 3.3.2 Discriminator

The defined discriminator is a simple Fully Convolutional Network (FCN), composed of 6 convolutional layers, that output a 2D matrix containing probabilities of the generated image being real. This model is presented in Fig. 3.3. As shown, the discriminator receives two input images which are the degraded image and its clean version (ground truth or cleaned by the generator). Those images were concatenated together in a  $256 \times 256 \times 2$  shape tensor. Then, the obtained volume propagated in the model to end up in a  $16 \times 16 \times 1$  matrix in the last layer. This matrix contains probabilities that should be, to the discriminator, close to 1 if the clean image represents the ground truth. If it is generated by the generator the probabilities should be close to 0. Therefore, the last layer takes a sigmoid as an activation function. After completing training, this discriminator is no longer used. Given a degraded image, we only use the generative network to enhance it. But, this discriminator shall force the generator during training to produce better results.

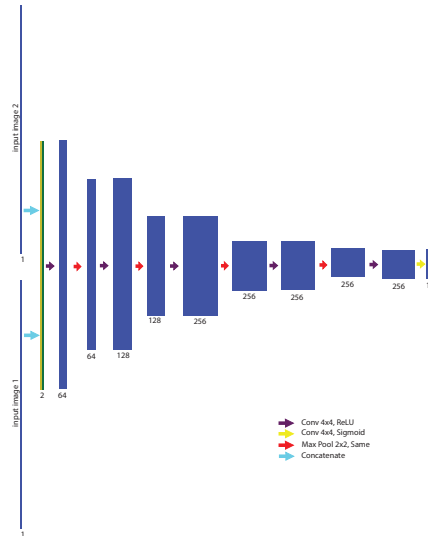


Figure 3.3: The Discriminator architecture

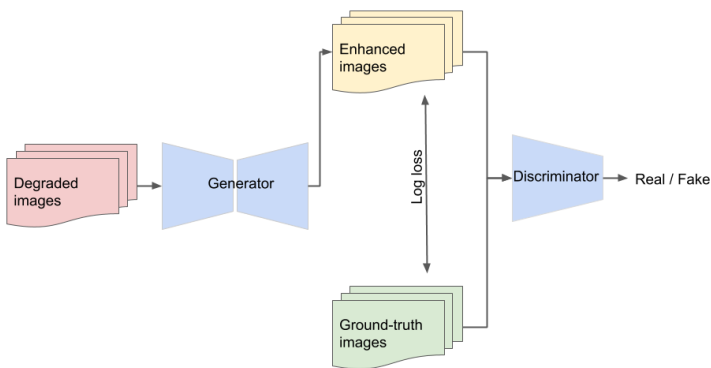


Figure 3.4: The proposed DE-GAN

### 3.3.3 Training process

Training our DE-GAN was as follows, we took patches from the degraded images of size  $256 \times 256$  and fed them as an input to the generator. The produced images are fed to the discriminator with the ground truth patches and the degraded ones. Then, as presented in equation 3.3, the discriminator starts forcing the generator to produce

outputs that cannot be distinguished from “real” images, while doing his best at detecting the generator’s “fakes”. This training is illustrated in Fig. 3.4 and it is done using Adam with a learning rate of  $1e^{-4}$  as an optimizer.

## 3.4 Experiments and results

### 3.4.1 Document cleaning and binarization

We begin our experiments with document cleaning. For this task, the Noisy Office Database which contains different types of degradation, and is presented in [203], is used. We defined 112 images for training and 32 for testing. From the 112 training images, a set of overlapped patches of size  $256 \times 256$  pixels were extracted. This has generated 1356 pairs of patches that were fed to our model. This first test intends to demonstrate the adversarial training effect. Thus, we train another model which is a simple FCN which is the U-net presented in Fig. 3.2. A validation set of 15 % from the training images was used in this model. The results obtained by both models are presented in Table 3.1. As could be interpreted, the result of the encoder-decoder network (U-net) is acceptable for denoising and cleaning tasks. But, our DE-GAN is further improving the results. Which exposes the reason for using adversarial training for these types of problems. For more comparison, we have participated in the Kaggle competition on denoising dirty documents <sup>1</sup>, we obtain a root mean squared error score of 0.01952. This makes our method one of the best approaches on the leaderboard.

Table 3.1: The obtained results of document cleaning using Noisy office database [203]

Model	SSIM	PSNR
FCN (U-net)	0.9970	36.02
<b>DE-GAN</b>	<b>0.9986</b>	<b>38.12</b>

Next, we compare our approach with state-of-the-art results in the document binarization problem. We take the DIBCO 2013 Dataset [149] for testing. While training our model was done with different versions of DIBCO Databases [64, 147, 148, 137, 151, 152]. Same as the previous test, a set of 6824 training pairs (patches of size  $256 \times 256$ ) was taken from its 80 total images. The obtained results are compared with several approaches in Table 3.2. Out of the results, we can say that DE-GAN is superior to the current state-of-the-art methods according to the following metrics [149]: Peak signal-to-noise ratio (PSNR), F-measure, pseudo-F-measure ( $F_{ps}$ ) and Distance reciprocal distortion metric (DRD). Some examples of DIBCO 2013 images binarization by DE-GAN are presented in Fig. 3.5.

<sup>1</sup><https://www.kaggle.com/c/denoising-dirty-documents/>

Table 3.2: Results of image binarization on DIBCO 2013 Database.

Model	PSNR	F-measure	$F_{ps}$	DRD
Otsu[138]	16.6	83.9	86.5	11.0
Niblack[135]	13.6	72.8	72.2	13.6
Sauvola et al.[164]	16.9	85.0	89.8	7.6
Gatos et al. [63]	17.1	83.4	87.0	9.5
Su et al. [177]	19.6	87.7	88.3	4.2
Tensmeyer et al [183]	20.7	93.1	96.8	2.2
Xiong et al. [197]	21.3	93.5	94.4	2.7
Vo et al. [188]	21.4	94.4	96.0	1.8
Howe [79]	21.3	91.3	91.7	3.2
<b>DE-GAN</b>	<b>24.9</b>	<b>99.5</b>	<b>99.7</b>	<b>1.1</b>

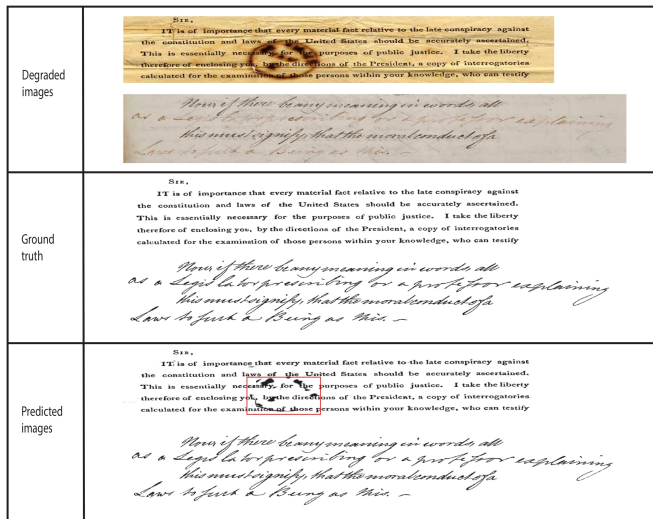


Figure 3.5: Binarization of degraded documents by DE-GAN, the result is satisfactory, except in some parts that were highly dense (the red boxes in the row of the predicted image)

To reflect the results of the previous Table, illustrative comparisons between those different methods could be found in Fig. 3.6 and Fig. 3.7. It is easy to visualize the superiority of our method over the classic methods, like those of [138, 135, 164], which fail to remove the background degradation from the document when it gets very dense because they are based on thresholds that make the degraded pixels classified as a text, or classifying the text pixels as damage to be removed. The recent approaches [79, 188], yield a better result than the classic ones and separate the text from the background successfully. However, our method gives a higher performance in terms of closeness to



the ground truth image.

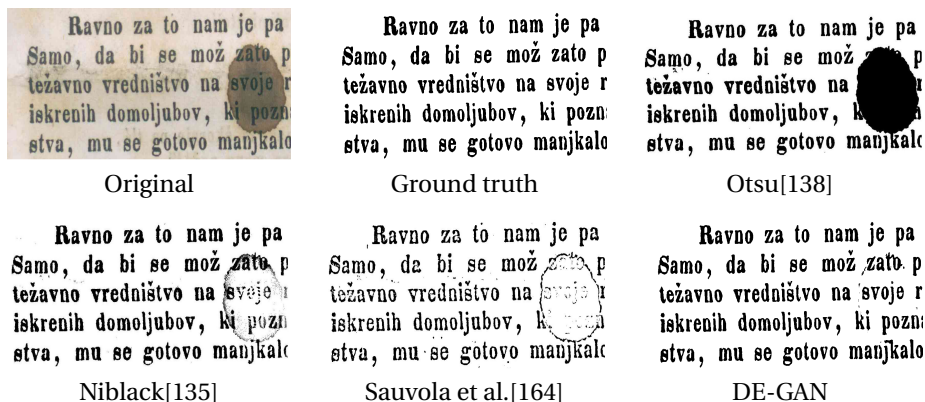


Figure 3.6: Qualitative binarization results produced by different methods of a part from the sample (PR5), which is included in DIBCO 2013 dataset

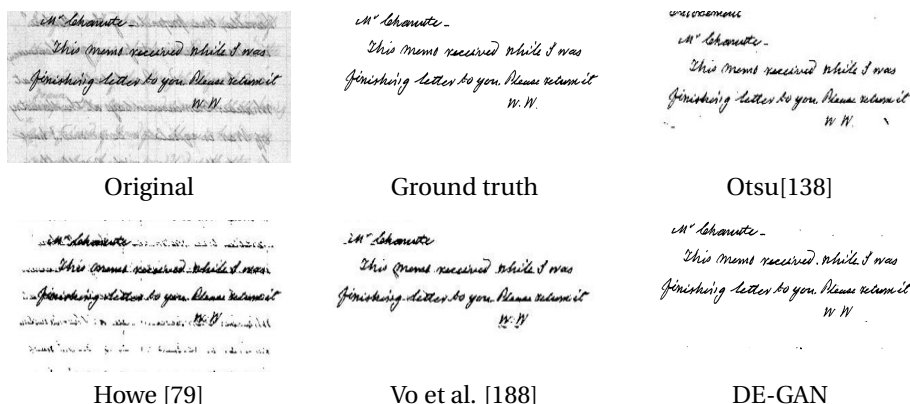


Figure 3.7: Qualitative binarization results produced by different methods of a part from the sample (HW5), which is included in DIBCO 2013 dataset

Moreover, we tested DE-GAN on a recent DIBCO dataset, which is DIBCO 2017 [152]. We train our model on 6098 patches from similar datasets [149, 64, 147, 148, 137, 151]. The comparison is done with the top 5 ranked approaches in ICDAR 2017 competition on document image Binarization [152]. 18 research groups have participated in the competition with 26 distinct algorithms. The results are presented in table 3.3, where you can notice the superiority of our DE-GAN over the different methods. It is to note that most of these approaches are based on encoder-decoder models and the

winner team was using a U-net with several data augmentation techniques. However, GANs were not exploited in this competition.

Table 3.3: Results of image binarization on DIBCO 2017 Database, a comparison with DIBCO 2017 competitors approaches.

Model	PSNR	F-measure	$F_{ps}$	DRD	Rank in the competition
10 [152]	18.28	91.04	92.86	3.40	1
17a [152]	17.58	89.67	91.03	4.35	2
12 [152]	17.61	89.42	91.52	3.56	3
1b [152]	17.53	86.05	90.25	4.52	4
1a [152]	17.07	83.76	90.35	4.33	5
<b>DE-GAN</b>	<b>18.74</b>	<b>97.91</b>	<b>98.23</b>	<b>3.01</b>	-

In addition, we compared our model with the most recent approaches, presented in the H-DIBCO 2018 competition [150] that was held at ICFHR 2018 conference. The results are presented in Table 3.4. As shown, our approach has the best performance on DIBCO 2017 test set and gives the second best DRD, PSNR, F-measure, and pseudo F-Measure on H-DIBCO 2018 test set. We note that the winner system in the competition integrates a lot of pre-processing and post-processing steps in their algorithm, which makes it more efficient for this particular H-DIBCO 2018 dataset. On the contrary, we are presenting a simple end-to-end model that shows a good ability in several datasets and enhancements tasks without any additional processing step. Finally, for more practical usage of the model, we tried to binarize some real (naturally degraded) documents as well, the degradation consists in stains and show-through. The obtained results are given in Fig. 3.8, the model is producing a better version of the real images, which will certainly improve their recognition rate.

Table 3.4: Results of image binarization on DIBCO 2017 and DIBCO 2018 Databases, a comparison with DIBCO 2018 competitors approaches.

Model	DIBCO 2018				DIBCO 2017				Rank in the competition
	PSNR	F-measure	$F_{ps}$	DRD	PSNR	F-measure	$F_{ps}$	DRD	
1 [150]	<b>19.11</b>	<b>88.34</b>	<b>90.24</b>	<b>4.92</b>	17.99	89.37	90.17	5.51	1
7 [150]	14.62	73.45	75.94	26.24	15.72	84.36	87.34	7.56	2
2 [150]	13.58	70.04	74.68	17.45	14.04	79.41	82.62	10.70	3
3b [150]	13.57	64.52	68.29	16.67	15.28	82.43	86.74	6.97	4
6 [150]	11.79	46.35	51.39	24.56	15.38	80.75	87.24	6.22	5
DE-GAN	16.16	77.59	85.74	7.93	<b>18.74</b>	<b>97.91</b>	<b>98.23</b>	<b>3.01</b>	-

### 3.4.2 Watermark removal

After testing our model in document cleaning and binarization, we will evaluate it on the problem of watermark removal. Dense watermarks (or stamps) can cause a huge

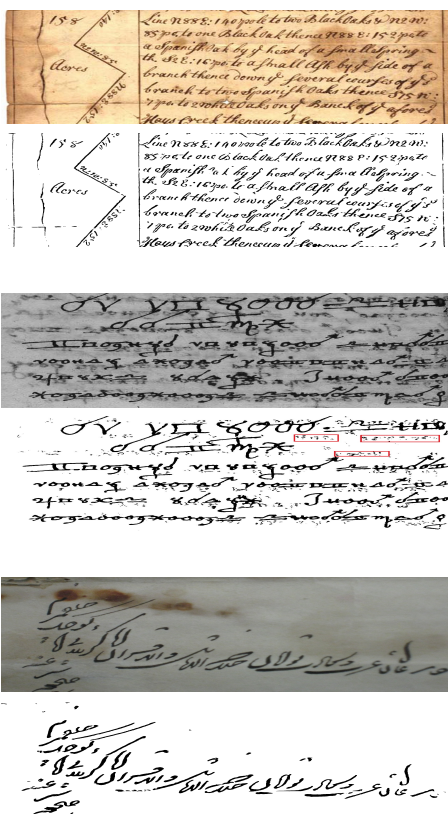


Figure 3.8: Binarization of three historical degraded documents by DE-GAN, the binarized version is presented under each original image. Some parts are not well recovered as shown in the red boxes.

deterioration in the *foreground* of the document, which makes it hard to be read. However, this problem was not investigated by the document analysis community. We decided to be the first that addresses it using DE-GAN. Hence, it was not possible to find a public dataset for testing. We created our own database which contains 1000 pairs (an image of a document with a dense watermark and stamps and its clean version).

The used watermarks have random texts, sizes, colors, fonts, opacities, and locations (see Fig. 3.9). As shown, these watermarks are sometimes covering the entire text making it unseen by the unaided eye. The code used to produce this data is available at GitHub<sup>2</sup>, for the same dataset used in our study the reader can contact the first author to obtain it. Training our DE-GAN was done, same as document cleaning, by using

<sup>2</sup><https://github.com/dali92002/watermarking-documents/blob/master/Watermarking.ipynb>

overlapped patches (7658 pairs of patches from 800 watermarked document images). While taking 200 documents for testing. Since, to the best of our knowledge, there is no approach in the literature that addresses this problem in documents. Comparing our obtained results was done with the approaches used in natural image watermark removal. The comparison results are presented in Table 3.5.

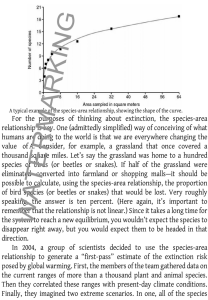


Figure 3.9: 4 Samples from our developed Dataset

Despite that, the watermarks used in our study were very dense and we believe that removing them is harder than the related approaches presented in Table 3.5. Our approach surpasses, by far, those in natural images. Fig. 3.10 shows some examples of watermark removal by DE-GAN, the produced images are preserving the text quality while removing the foreground watermarks. In addition, since the presented watermarked documents were synthetically made, it was interesting to apply DE-GAN to remove watermarks from a naturally degraded document. Fig. 3.11 shows that DE-GAN successfully removes a dense watermark from a document paper. As you can see, the watermark is completely removed, and the reader or the OCR system can easily read the enhanced document compared to the degraded one.

Table 3.5: Results of watermark removal

Model	PSNR	SSIM
Dekel et al. [49]	36.02	0.924
Wu et al. [194]	23.37	0.884
Cheng et al. [38]	30.86	0.914
<b>DE-GAN</b>	<b>40.98</b>	<b>0.998</b>

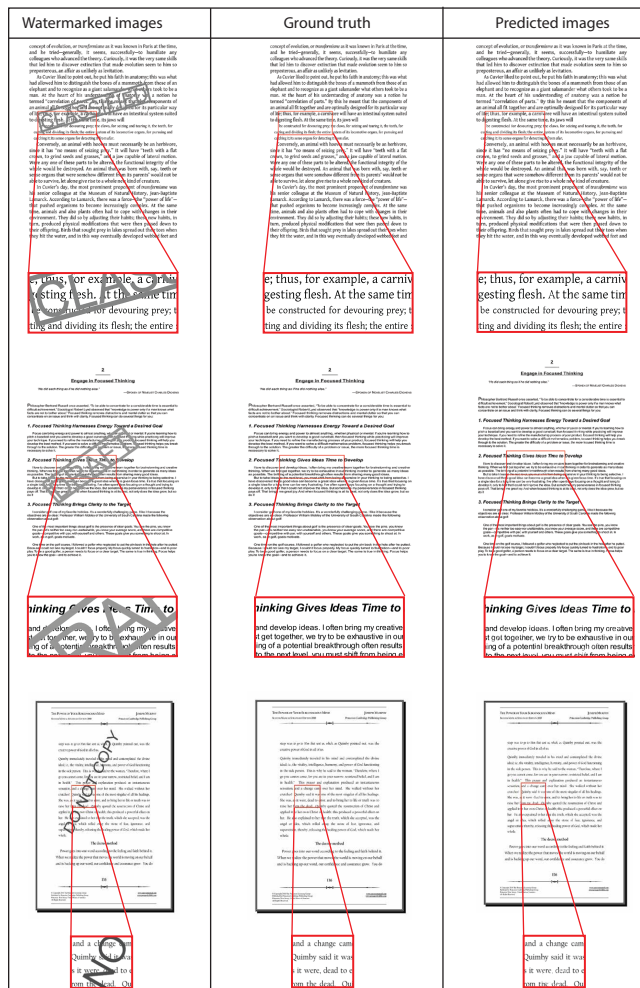


Figure 3.10: Watermark removal by DE-GAN

### 3.4.3 Comparison with other GAN models

As it is a fact that our model is inspired by the pix2pix model [85] (we are using a deeper generator and a different additional loss), it would be useful if we tried some other similar models that are based on GANs and dedicated to the same image-to-image translation problem. For this aim, cycleGAN [214] and pix2pix-HD [191] models are considered for the comparison. We evaluate these models on H-DIBCO 2018 dataset [150] with the same conditions and data used to train the DE-GAN. The quantitative and qualitative obtained results are presented in Table 3.6 and Fig. 3.12, re-



Figure 3.11: Qualitative results for dense watermark removal. Above, is a section from watermarked invoice. Below, it's enhanced version. Some parts of the text in the invoice were blurred due to privacy constraints. Because of different domains, synthetic vs real, we can see that some tiny parts of the watermark were not completely removed (red boxes).

spectively. Experimental results show the superiority of DE-GAN compared to cycleGAN and pix2pix-HD in achieving higher PSNR, F-measure, and Fps and a lower DRD. We note that the unsupervised training capabilities of CycleGAN are quite useful since paired data is harder to find in document enhancement applications. For pix2pix-HD, the results are promising, since the training samples that we used for training were few (the number of DIBCO samples is small if we split them into patches with size  $512 \times 1024$ , that's why we used some flips of images to augment the data). With more data, we believe that pix2pix-HD could perform much better.

Table 3.6: Results of image binarization for DIBCO 2018 Database

Model	PSNR	F-measure	$F_{ps}$	DRD
cycleGAN	11.00	56.33	58.07	30.07
pix2pix-HD	14.42	72.79	76.28	15.13
<b>DE-GAN</b>	<b>16.16</b>	<b>77.59</b>	<b>85.74</b>	<b>7.93</b>

### 3.4.4 Document deblurring

The DE-GAN model presented in this Chapter is able to outperform many state-of-the-art approaches in different problems like binarization, denoising, and watermark removal. To experimentally prove the efficiency and flexibility of the proposed method, we evaluate it in a more challenging scenario, which is document deblurring. We use 4000 patches from the dataset developed in [80] to train our model, and 932 patches for testing. Noting that, in [80] a convolutional neural network architecture is proposed to address the problem. Thus, we will compare the results with this CNN and pix2pix-HD models trained on this selected data. The obtained results are presented in Table 3.7. We can see that GAN's models surpass CNN. This is much clear in the qualitative results of some patches presented in Fig 3.13. We can also see that DE-GAN gives similar

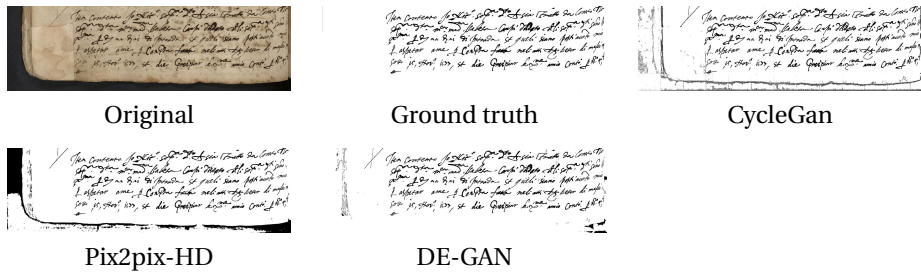


Figure 3.12: Qualitative binarization results produced by different models of the sample (9) from H-DIBCO 2018 dataset

results to pix2pix-HD, however, it is more accurate for predicting some characters. For example, in the second patch row, third line, the word "kind" is correctly predicted by DE-GAN but it is predicted as "bind" by pix2pix-HD. We note that the used dataset is composed of 300x300px image patches, which can explain why pix2pix-HD does not give better performance (it works generally with a larger input patches with a size of 512x1024, or 1024x2048).

Table 3.7: The obtained results of document deblurring

Method	PSNR
CNN [80]	19.36
pix2pix-HD [191]	19.89
<b>DE-GAN</b>	<b>20.37</b>

### 3.4.5 OCR evaluation

After the quantitatively and qualitatively evaluation of the resulting enhanced images presented previously, we compare in what follows the performance of OCR on degraded and enhanced documents. For this aim, we took a set of 4 images (2 degraded ones from DIBCO datasets, and 2 images with a dense watermark from our dataset). Then, we used Tesseract OCR [168] to recognize those images and their enhanced versions with DE-GAN. We found that the proposed enhancement method boosts the baseline OCR performance by a large margin, and the character error rate is decreased from 0.37 for the degraded documents to 0.01 for the enhanced ones. Fig. 3.14 shows a tiny example of this process. In each row, you can find a line of a degraded document image and the text produced by the OCR system, then its enhanced version followed the OCR text.

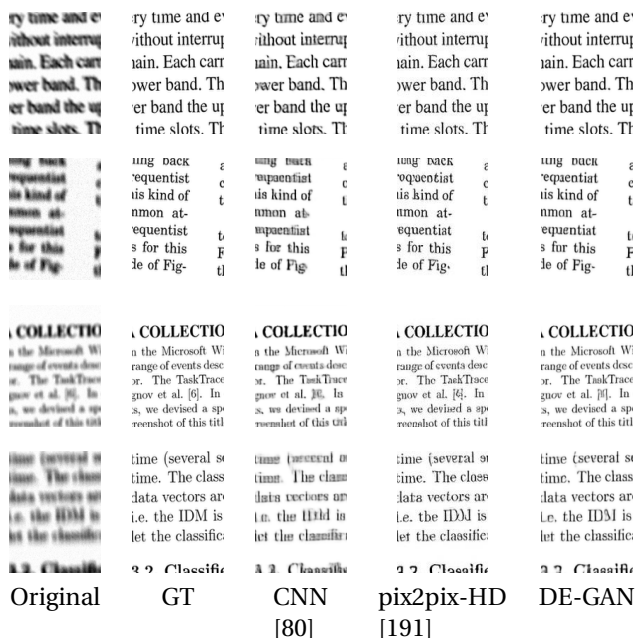


Figure 3.13: Qualitative deblurring results of some patches produced by different methods

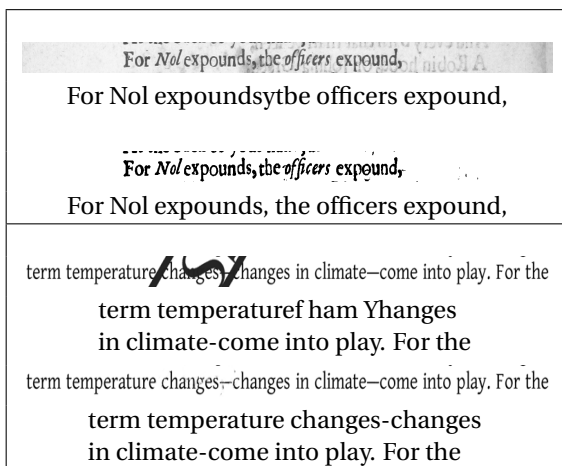


Figure 3.14: Qualitative results for Tesseract recognition of some text lines



## 3.5 Conclusion

In this Chapter, we proposed a Document Enhancement Generative Adversarial Network named DE-GAN to restore severely degraded document images. To the best of our knowledge, this was the first application of GANs for studying document enhancement problems. Moreover, we present a new problem in document enhancement which is dense watermark (or stamps) removal, hoping that it takes the attention of the document analysis community. Extensive experiments show that DE-GAN achieved interesting results in different document enhancement tasks that outperform the fully convolutional networks, cycleGAN, and pix2pix-HD models. Furthermore, we achieve improved results compared to many recent state-of-the-art methods on benchmarking datasets like DIBCO 2013, DIBCO 2017 and H-DIBCO 2018.

We showed that the proposed enhancement method boosts the baseline OCR performance by a large margin. Hence, in the next Chapter, we will add the OCR/HTR evaluation in the discriminator part. Thus, we can give the discriminator the ability to read the text to decide if it is real or fake, which will force it to generate more readable images. This will be done in the next Chapter.

# Chapter 4

## A Multi-Task Adversarial Network for Handwritten Document Image Enhancement

---

*In this Chapter, we propose an end-to-end architecture based on Generative Adversarial Networks (GANs) to recover the degraded documents into a clean and readable form. Unlike the most well-known document binarization methods, which try to improve the visual quality of the degraded document, the proposed architecture integrates a handwritten text recognizer that promotes the generated document image to be more readable. To the best of our knowledge, this is the first work to use text information while binarizing handwritten documents. Extensive experiments conducted on degraded Arabic and Latin handwritten documents demonstrate the usefulness of integrating the recognizer within the GAN architecture, which improves both the visual quality and the readability of the degraded document images. Moreover, we outperform the state of the art in H-DIBCO challenges, after fine-tuning our pre-trained model with synthetically degraded Latin handwritten images, on this task.*

---

### 4.1 Introduction

As we said in the previous Chapter, one of the problems that Handwritten Text Recognition (HTR) systems are facing is the degradation of an inputted document. This significantly decreases the reading performance, reflecting on its utility. Indeed, many

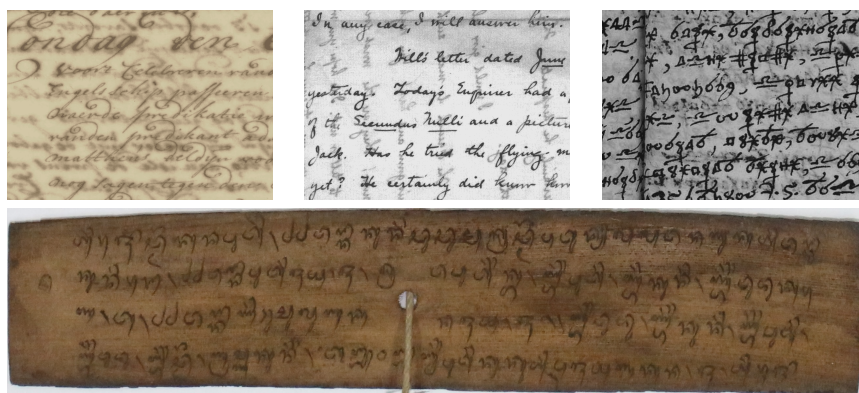


Figure 4.1: Examples of the degradation that can be appeared in handwritten text images.

degradation scenarios can be attached to a handwritten document, especially historical ones. Degradation includes background noise, corrupted text, dust, wrinkles, and historical effects just to name a few related to the condition of the document itself [146]. The bad scanning process can also produce problems (shadows, blur, light distortion, angle, etc.) [175, 189]. Moreover, some documents contain watermarks or stamps inserted for security reasons, those can cover the text and obstruct the HTR engine [174]. Some degradation examples are presented in Figure 4.1, as it can be seen, cleaning the document before passing it to the HTR stage should be done.

This cleaning task, called document enhancement, includes different recovering techniques, to reverse the degradation effect, for example, Binarization, dewarping, deblurring, watermark removal, etc. Classic recovery techniques integrate image processing algorithms to be used as a filter that separates the degradation from the text. However, those methods are failing in removing the high degradation. Also, their parameters are usually set depending on the quality state of the addressed document to produce an optimal result. Thus, manual intervention is needed in some cases to adjust the parameters, which is quite costly.

Given this, some modern document recovery techniques are appearing, using machine learning tools. Those are training deep learning models, mainly Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), to learn the parameters for a direct mapping of any degraded document image into a clean binary version (without a restriction on degradation level) [181, 96, 211]. Similar to those, we proposed in [174] a document enhancement model called DE-GAN. However, despite the high accuracy that we achieved in various enhancement tasks (Binarization, cleaning, watermark removal, deblurring), an important evaluation was not done. In fact, the goal of document enhancement is to provide a cleaner version of the image which is highly beneficial for HTR engines. But, in the mentioned approaches (includ-

ing ours), the evaluation was conducted using only the visual similarity measurement between the recovered image and the Ground Truth (GT) clean version with some metrics that depend on pixel values. Thus, an HTR evaluation (which means, passing the images to an HTR engine and comparing the recognized text with the GT) is missing, for a better validation of the developed approaches. Also, these models are generally trained using only the images, while ignoring the text. As result, a model can easily evolve to deteriorate the text while cleaning the degraded image.

Motivated by those challenges, we propose a new method consisting in an improved version of our previously developed DE-GAN, which was designed to recover the handwritten document to a clean version while ensuring its readability. Our approach is a deep learning model based on GANs that learns its parameters not only from the handwritten image pairs (degraded + GT) but also from the associated GT text. For this aim, we propose to add a recognizer that is trained jointly in a GAN model to assess the readability of the recovered document image. Hence, the model shall learn the best mapping of the degraded image to be as clean as possible while keeping its text readable. To accomplish this, and since the used datasets for document binarization do not (or rarely) contain the text information, we used two publicly available handwriting text images datasets (KHATT for Arabic script and IAM for Latin script) that are originally used for HTR to create degraded versions from the GT clean text lines images. The contribution of this Chapter can be summarized as follows:

- To the best of our knowledge, this is the first work that integrates a recognition stage in a document binarization model. Thus, the degraded handwritten document will be recovered while maximizing its readability, simultaneously. This is done by combining the GAN and the Connectionist Temporal Classification (CTC) losses functions: We eliminate the noise while preserving the handwritten text strokes.
- We demonstrate that training the recognizer progressively (on images ordered from the degraded domain to the clean versions), improves the recognition performance.
- The proposed model is simple and flexible to restore different forms of degradation, independently of the document language. This was shown by the experiments conducted on two created datasets namely degraded-IAM (Latin script) and degraded-KHATT (Arabic script).
- We achieved the SOTA performance in handwritten document binarization according to H-DIBCO benchmarks.

The rest of this Chapter is organized as follows. We present our proposed model in Section 4.2. After that, experimental results and comparisons with recent methods will be described in Section 4.3. Finally, a conclusion with a future direction is given in Section 4.4.

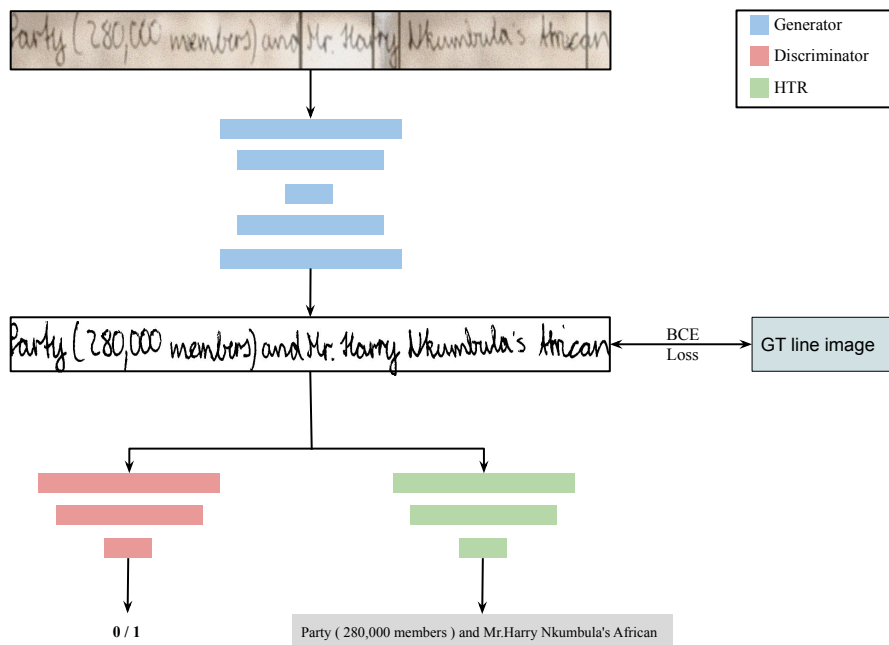


Figure 4.2: Proposed architecture for document binarization.

## 4.2 Proposed Method

We treat recovering a clean version from a handwritten degraded document as an image-to-image translation task using a generative model. Our GAN architecture is composed of three main parts, as shown in Figure 5.1: A generator, a discriminator, and a handwritten text recognizer. Since we are using the text information, the patches that are used during training should be in a readable form by an HTR, after binarization. Thus, the model is designed to be working at the handwritten line images level. During training, the generator is conditioned on the degraded line image to generate the clean version. The generated image is passed to the discriminator to assess it as real (looks clean) or fake (looks degraded), for ensuring a realistic visual recovery. The image is also passed also to the HTR model to read it and compare the recognized text to the GT, hence, maintaining its readability while recovering it. The discriminator, as well as the recognizer, passed their feedback about the generated image through the adversarial loss. Noting that another additional Binary Cross Entropy (BCE) loss is integrated into the generator, for faster convergence. In this way, the generator parameters are learned to produce a handwritten image that is as clean as possible, while keeping the text quality. In what follows, we explain the three components presented in our archi-

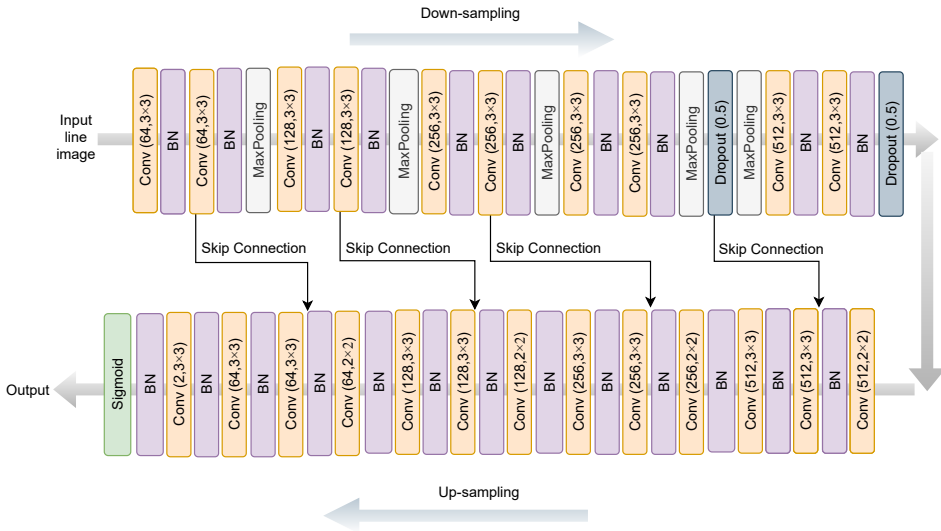


Figure 4.3: Generator's architecture design used in this study.

texture in more detail.

## 4.2.1 Generator

Since we are doing an image-to-image translation process, the generator is designed as an auto-encoder model. We employ the U-net [157] for this task, in which the inputted image is encoded through a sequence of convolution layers with a down-sampling to reach a specific layer. After that, the image is decoded with a sequence of up convolution layers with an up-sampling. The model involves some skip connections after every two successive layers to recover images with a lower deterioration since the goal is to keep the text while removing the degradation. Thus, skip connections can help the decoder in maintaining the text features while producing the image. Figure 4.3 shows further details about the used generator. As can be seen, it is composed of 23 convolutional layers, with Dropout regularization and batch normalization layers. The output of this model is a single channel (in gray level) image, assumed to be the cleaned version of the inputted degraded image.

## 4.2.2 Discriminator

The discriminator is another Fully Convolutional Network (FCN) that produces an assessment of the generated image in terms of visual similarity (pixel level) with the GT

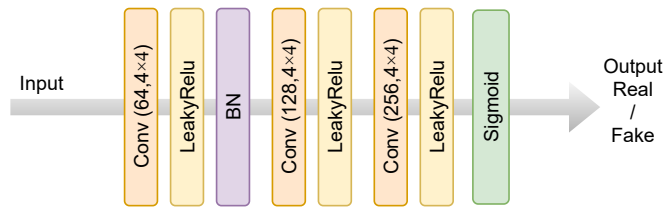


Figure 4.4: Discriminator's architecture used in this study.

(real) images. The model was designed to take a degraded image with its clean version and output the class "real" if the clean version is the real GT, or assign the class "fake" if the clean image was produced by the generator. Both input images, which have of course the same size, are concatenated in an  $H \times W \times 2$  shape. Then, the obtained volume is propagated in the discriminator model detailed in Figure 4.4, to end up in the last layer as a form of  $H/16 \times W/16 \times 1$  matrix. During training, this matrix contains values that are equal to 1 in case of inputting the GT as a clean image, and equal to 0 in case of inputting the generator-based enhanced image.

### 4.2.3 Handwritten Text Line Recognizer

The used handwritten recognizer is a Convolutional Recurrent Neural Network (CRNN) model, following the architecture presented recently in [134] and considered among the best HTR architectures. Noting that, any other HTR can be also used for this task, for instance: [179, 90, 93]. The model architecture is detailed in Figure 4.5. After enhancing the image by the generator it is inputted to an encoding stage that uses convolutional and gated convolutional layers, with integrated regularization techniques. The encoded image is passed later to the decoding stage, which consists in two bidirectional Gated Recurrent Unit (GRU) layers. Finally, the CTC is used to decode the feature frames into text characters. The CTC layer is having the size of the character set plus one additional symbol corresponding to the blank symbol. During training, the recognizer could be fitted with two types of clean images, forming the following two scenarios:

- **S1:** The Recognizer is trained at each iteration with the GT images that are related to the degraded batch images inputted to the generator. The GT images are used with the associated GT text transcription for training in this process.
- **S2:** The Recognizer is trained using the images enhanced by the generator at each iteration with the associated GT text. The intuition behind this is that we may obtain a better recognition convergence that is going progressively from the degraded domain to the clean domain.

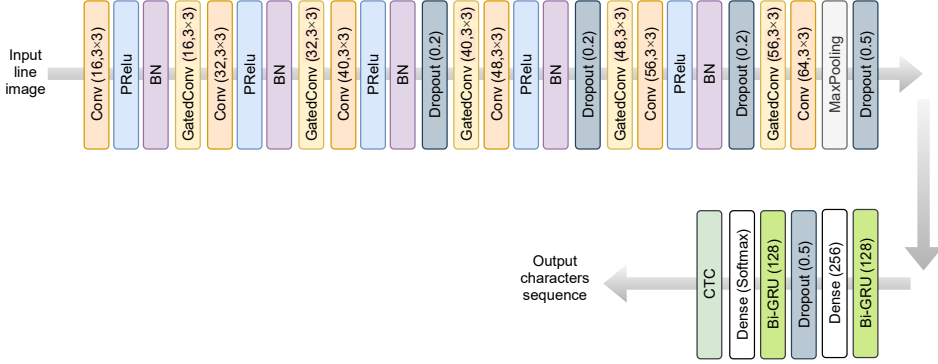


Figure 4.5: Workflow of the CNN-Bi-GRU recognizer's architecture.

#### 4.2.4 Training process

The different components presented above were trained jointly. The generator  $G$ , which is a function having the parameters  $\theta_G$ , is conditioned on the degraded image  $I_d$  to provide its cleaned image that should be as close as possible to the GT image  $I_{gt}$ . This image is passed to be validated by the discriminator  $D$  and handwritten recognizer  $R$ , with parameters  $\theta_D$  and  $\theta_R$ , respectively.  $D$  is giving an assessment of the cleaned image about its cleanliness to be Real or Fake,  $P(\text{Real}) = D_{\theta_D}(G_{\theta_G}(I_d))$ . This adversarial training process of  $G$  and  $D$  can be formalized by:

$$\mathcal{L}_{adv}(\theta_G, \theta_D) = \mathbb{E}_{I_d, I_{gt}} \log[D_{\theta_D}(I_d, I_{gt})] + \mathbb{E}_{I_d} \log[1 - D_{\theta_D}(I_d, G_{\theta_G}(I_d))] \quad (4.1)$$

$R$  is recognizing the generated image to maintain its readability with the CTC decoder,  $CTC(t, R_{\theta_R}(G_{\theta_G}(I_d)))$ , where  $t$  is the GT text. Note that it is trained with a clean version of the image (whether using S1 or S2, presented above), at each same iteration:  $CTC(t, R_{\theta_R}(I_{gt}))$ . Also, for faster convergence, a simple BCE loss is used in the generator between the cleaned images and the GT ones,  $BCE(\theta_G)$ . Thus, the generator is being affected by three factors to produce its generation. The whole architecture is formalized as:

$$\mathcal{L}(\theta_G, \theta_D, \theta_R) = \min_{\theta_G} \max_{\theta_D} \mathcal{L}_{adv}(\theta_G, \theta_D) + \lambda(E_{t, I_d} CTC(t, R_{\theta_R}(G_{\theta_G}(I_d))) + \beta BCE(\theta_G) \quad (4.2)$$

Where  $\lambda$  and  $\beta$  are the weights balancing the components intervention to produce the final generated image. During our experiments, we set  $\lambda$  to 1 and  $\beta$  to 10. For training, we used Adam's optimizer for the generator and discriminator components, while using the RMSProp for the handwritten text recognizer.



## 4.3 Experiments and Results

We provide in this Section the experiments that were done to validate the effectiveness of our proposed method. First, we start by presenting the metrics and datasets used in our evaluation.

### 4.3.1 Metrics

Following the usual approaches for handwritten document image binarization [148], we use the same metrics to validate the cleaned images (Same as in the previous Chapter). Those metrics which compare the image's visual similarity with the GT clean ones are Peak signal-to-noise ratio (PSNR), F-Measure (FM), pseudo-F-measure (Fps), and Distance reciprocal distortion metric (DRD). In addition, since we are using the text information to validate our model, we utilize as well the HTR metrics for comparing the recognized text to the GT one. These metrics are based on the Levenshtein distance [115], and they consist of the Character Error Rate (CER) and the Word Error Rate (WER) measures.

### 4.3.2 Handwritten text databases

Usual handwritten document binarization databases do not contain text information [146, 26, 9]. Thus, we opt to create synthetically degraded images from the databases used in HTR tasks in order to exploit the GT text provided within these datasets. We address in this study two different alphabets: Arabic and Latin. From each, we took the most used database for handwritten text line image recognition: KHATT [125] and IAM [127], to add degradation. We call the created datasets, degraded-KHATT, and degraded-IAM.

#### Degraded-KHATT

The KHATT dataset was developed for Arabic manuscript recognition and contains text line images with their associated GT texts. In our experiments, we used 6161 lines for training and a set of 1861 lines for testing, while a set of 940 lines was used for validation as it was done in [179]. Then, we added random distortions as shown in Figure 4.6 to obtain the degraded-KHATT dataset. To accomplish this, we insert different background images containing some flaws or artifacts. These background images are extracted mainly from public historical documents such as Nabuco, Bickley diary, and Persian datasets. We have also applied different distortion operations, especially, dilation, erosion and blurs using random kernel sizes ( $2 \times 2$  and  $3 \times 3$  for dilation,  $2 \times 2$ ,  $3 \times 3$  and  $4 \times 4$  for erosion and from  $1 \times 1$  to  $15 \times 15$  for blurring). We inserted also random vertical lines having random widths in order to simulate the noise that can occur in old historical documents.



Figure 4.6: Examples of distorted line images of the degraded-KHATT database used in this study, images are presented in gray level.

### Degraded-IAM

The IAM dataset was proposed for handwritten Latin script text recognition. It contains 8962 line images taken from the Lancaster-Oslo/Bergen (LOB) corpus. To insert degradation, we used the same as in KHATT, 6161, 940, and 1861 line images for training, validation, and testing, respectively. We add dense backgrounds to simulate real historical deteriorated images same as it was done for the degraded-KHATT presented above. Examples of the obtained degraded-IAM are illustrated in Figure 4.7.

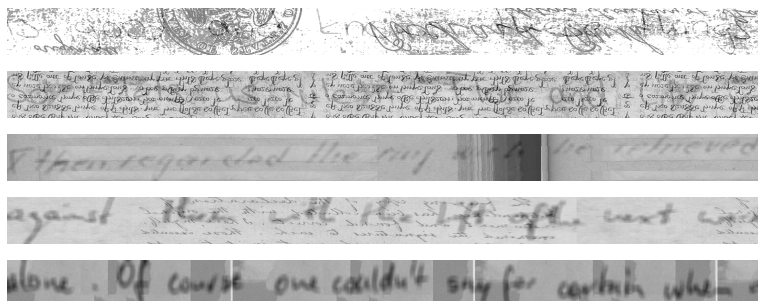


Figure 4.7: Examples of distorted line images of the degraded-IAM database used in this study, images are presented in gray level.

### 4.3.3 Results

#### Arabic handwritten texts images recovery

For Arabic, we fed the proposed model with the training set of the created degraded-KHATT database. As stated above, the generator is trained to map the degraded image into a clean version, which will be evaluated by the discriminator and the recognizer. It is to note that in our experiments (for Arabic and Latin manuscripts), we used a high degradation for a meaningful evaluation in the hard scenarios. Also, we separate the background types between the training and testing sets (i.e there is no intersection in the background noise between the two sets).

Table 4.1 illustrates the obtained results of the performed image binarization methods on the test set of the degraded-KHATT database. Reminding that we proposed two binarization scenarios sharing the same GAN architecture and integrating a CRNN recognizer, S1 and S2 stated before. The key difference between the two scenarios is the data fed to the recognizer during the GAN training stage. In scenario S1, the recognizer (we call it CRNN1) is fed with ground truth clean images, while it is fed with generated images (cleaned by the generator) in the second scenario (called CRNN2). As it can be seen, contrary to the previous related approaches, we evaluate the image in its visual quality (Binarization performance) and readability (recognition performance) at the same time. For readability, we tested each of our scenarios S1 and S2 (Reco. CRNN1 and Reco. CRNN2) using the two recognizers (CRNN1 and CRNN2).

To compare our approach, we used a simple GAN architecture as a baseline. The architecture contains the same generator and discriminator of our architecture, but, without using a recognizer. We compare also with the method presented in [175] for printed text recovery, where an OCR is used during training as a part of the discriminator. However, since we are doing a handwritten text recognition (not optical text). We modify it by training an HTR having the same architecture as [134] to use it as a part of

Table 4.1: Image binarization results for the *test set* (degraded-KHATT database). (A → B): The CRNN is trained on images from domain A and tested on images from domain B. Deg.: Degraded images. Reco.: Recognition performance.

Method	Binarization Performance (Visual Quality)				Reco. CRNN1 %		Reco. CRNN2 %	
	PSNR	FM	Fps	DRD	CER	WER	CER	WER
CRNN [134] (GT → GT)	ND	ND	ND	ND	12.04	32.39	-	-
CRNN [134] (Deg. → Deg.)	4.80	25.45	25.70	107.22	30.34	54.44	-	-
CRNN [134] (GT → Deg.)	4.80	25.45	25.70	107.22	91.18	100	-	-
Baseline cGAN	<b>15.52</b>	75.01	75.11	<b>6.05</b>	29.24	53.68	-	-
cGAN [175]	15.10	75.56	75.75	11.78	28.84	54.37	-	-
<b>Ours (S1)</b>	15.45	<b>77.45</b>	<b>77.60</b>	7.97	27.03	52.84	<b>24.33</b>	<b>47.67</b>
<b>Ours (S2)</b>	15.44	74.52	74.62	6.18	27.90	53.49	25.31	48.48

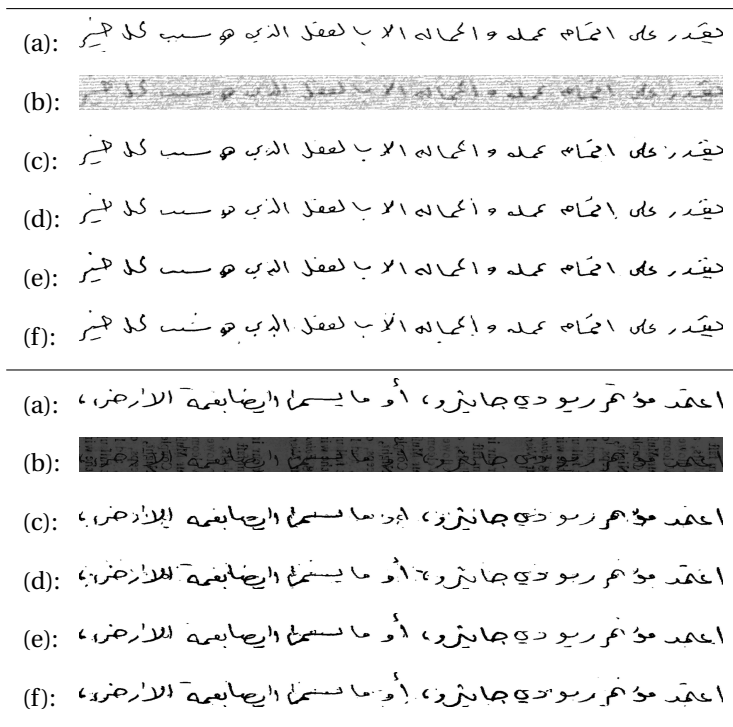


Figure 4.8: Results of our proposed method for recovering degraded lines images. (a): GT, (b): Distorted, (c): Baseline cGAN, (d): cGAN [175], (e): Ours S1, (f): Ours S2.

the discriminator, more details are given in [175].

Out of the results, we can see that using the GT images, a trained HTR engine based on CRNN [134] is reaching a CER of 12.04 % and 32.39 % as a WER, this is considered as our upper bound for recognition. Using the same model to recognize the degraded images, we obtain a poor performance of 91.18 % in CER, obviously because the model is trained on clean data. If we train the model on the degraded train set, it results in 30.34 % of CER and 54.44 % of WER. This experiment is done to verify later if we can surpass this performance (as a baseline) by cleaning the images and then reading them, instead of training a model on the degraded domain.

The different binarization approaches, as can be noticed, are enhancing the visual quality and the readability of the degraded lines. However, we can see that the baseline cGAN which is not taking the text into consideration while cleaning the image, is producing a result having a better visual quality in terms of PSNR and DRD, but worse readability compared to the methods integrating the text information. For the recognizer-based methods, it is clear that our recovery method (S1) is leading to the best performance in terms of having a good visual enhancement while conserving text

readability. Since by recognizing the produced images, we get a CER of 27.03 % and a WER of 52.84 % when using the recognizer of S1 and a CER of 24.33 % and a WER of 47.67 % when using the S2 recognizer. This proves that using the text information during binarizing the images is useful. Also, we notice that the progressive learning of an HTR (training in order from the degraded images to their clean versions) in a multitask framework, is better for the recognition task. However, using the HTR pre-trained with the clean GT images (as a separate task) during enhancing the document, is better for the binarization performance (visually). To illustrate our method's effectiveness, we show in Figure 4.8 and Figure 4.9 some qualitative results of recovering the distorted lines, ranging from easy distortions to hard ones. We can see that our method is the most successful in producing clean images, especially in cases of highly degraded ones. In fact, it can even recover the vanished handwritten text strokes.

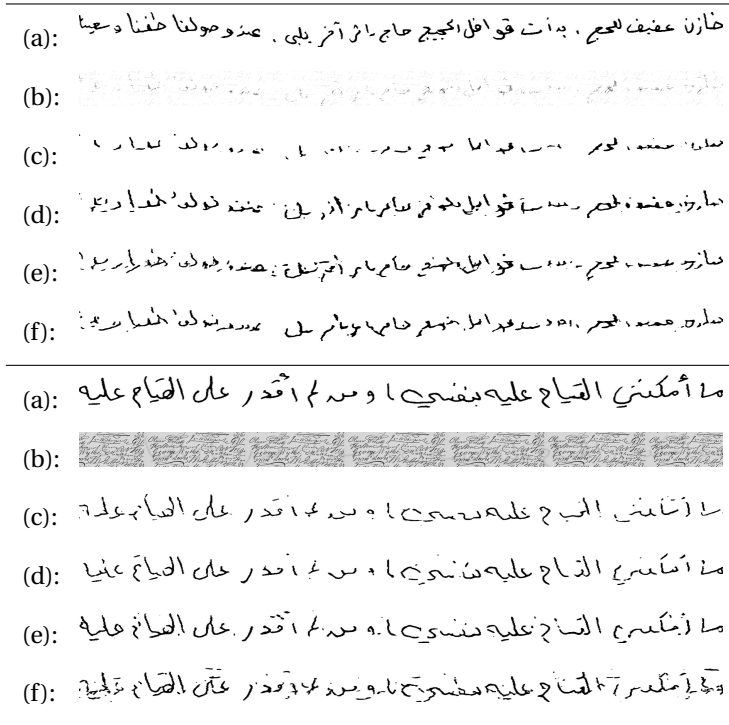


Figure 4.9: Results of our proposed method for recovering extremely degraded lines images. (a): GT, (b): Distorted, (c): Baseline cGAN, (d): cGAN [175], (e): Ours S1, (f): Ours S2.

Furthermore, as we stated above and since different weights can be used in the recognizer loss level to control the enhancement, we perform an ablation study to evaluate the right trade off between the visual quality and the readability during enhancement. In other words, the effectiveness of the weight  $\lambda$  presented in Equation 4.2. This

Table 4.2: Impact of the recognizer weight on the final generated image.

$\lambda$	Binarization Performance (Visual Quality)	Reco. performance	
	PSNR	CER%	WER%
0.5	<b>17.94</b>	11.98	31.07
<b>1</b>	17.88	<b>11.74</b>	<b>31.05</b>
5	17.71	12.66	32.44
10	17.07	15.22	36.74
20	16.32	19.72	40.84

is done by varying the weight  $\lambda$ , then training the model with that setting, and finally measuring the visual quality and readability (using the recognizer of S1) at each time to have the right option. The obtained results are shown in Table 4.2, where the experiments were carried out using the first scenario (S1) on a set of the Degraded-KHATT database, and ended up by selecting the setting of  $\lambda$  to be 1.

### Latin handwritten texts images recovery

For the Latin manuscript, we performed the same experiments using the degraded-IAM dataset. The obtained results are presented in Table 4.3. As it can be seen, training a CRNN [134] on the degraded images leads to 40.34 % as a CER and a WER of 74.05 %, with an obvious poor visual binarization quality since there was not a performed binarization with this way (using directly the degraded version). Contrary, by cleaning the image and passing it to the recognizer, better results were obtained. Here, same as the previous experiment, we are comparing our method to the basic GAN (without a recognizer), to validate the use of text information in our current method and our proposed one in [175]. It can be noticed that our method surpasses both GAN methods in visual quality, and achieves the best text recognition rate compared to the other options. By using the recognizer trained in S1, we boost the CER by 1.50 % compared to [175], 5.46 % compared to the basic GAN and 14.29 % compared to reading handwritten images in the distorted domain. Moreover, using the recognizer trained in S2 we can even improve the CER result by 4.07 %.

Furthermore, we show some qualitative results in Figures 4.10, 4.11 and 4.12, to visualize the performances of the different methods. Of course, reading the degraded image by a model trained on the GT clean images is not a suitable option. Also, training a model on degraded images is not improving the recognition, especially in hard scenarios. That is why, enhancing the image and then reading it is the better solution. As it can be seen, our method is better in this practice especially than the baseline cGAN (without a recognizer), because ours is a text conservative method. Hence, it maps the image to a clean but readable domain, while the basic GAN is mapping the image to a visually clean version, without taking the text into consideration (see Figures 4.11 and 4.12).

Table 4.3: Image binarization results for the *test set* (degraded-IAM database). (A  $\rightarrow$  B): The CRNN is trained on images from domain A and tested on images from domain B. Deg.: Degraded images. Reco.: Recognition performance.

Method	Binarization Performance (Visual Quality)				Reco. CRNN1 %		Reco. CRNN2 %	
	PSNR	FM	Fps	DRD	CER	WER	CER	WER
CRNN [134] (GT $\rightarrow$ GT)	ND	ND	ND	ND	11.92	36.07	-	-
CRNN [134] (Deg. $\rightarrow$ Deg.)	6.01	26.13	26.12	70.81	40.34	74.05	-	-
CRNN [134] (GT $\rightarrow$ Deg.)	6.01	26.13	26.12	70.81	90.46	99.50	-	-
Baseline cGAN	14.99	75.44	75.01	5.91	31.51	60.95	-	-
cGAN [175]	15.86	80.89	80.83	5.00	27.55	58.08	-	-
<b>Ours (S1)</b>	<b>15.97</b>	<b>81.69</b>	<b>81.55</b>	<b>4.83</b>	26.05	56.07	<b>21.98</b>	<b>49.74</b>
<b>Ours (S2)</b>	15.87	81.12	81.16	5.09	27.48	58.35	23.07	51.15

For the sake of more confirmation and to prove that our model is independent from the used recognizer, we took a different state-of-the-art HTR that is Puigcerver’s model [87] trained on the GT images of IAM and KHATT original datasets. Then, we carried a binarization stage to our degraded databases using different methods (including ours) and measure the final recognition performance. As it can be seen from Table 4.4, our proposed binarization method enhances the performance of the recognizer compared to the use of images binarized by the classic methods [138, 164] or the recent cGAN’s based one. Also, we can confirm the efficiency of our proposed binarization method compared to the baseline cGAN which did not integrate the text readability information.

Table 4.4: Impact of the proposed binarization method (scenario S1) on the recognition performance by a HTR system.

Dataset	Binarization Method	CER%	WER%
degraded-KHATT	Otsu [138]	54.28	85.42
	Sauvola [164]	58.42	99.57
	cGAN [175]	28.32	53.96
	Baseline cGAN	28.61	53.73
	<b>Ours (S1)</b>	<b>26.57</b>	<b>52.31</b>
degraded-IAM	Otsu [138]	62.62	81.96
	Sauvola [164]	72.48	98.00
	cGAN [175]	27.21	58.18
	Baseline cGAN	31.31	61.79
	<b>Ours (S1)</b>	<b>25.79</b>	<b>56.43</b>



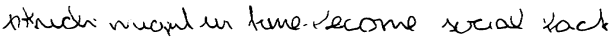
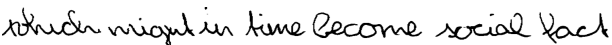


GT image:	
GT text:	which might in time become social fact
R(GT):	which might in time become social fact
Distorted:	
R(D):	pse nvgd in hame Cecome social hot
R(GT):	f
Baseline:	
R(GT):	ntauder muornd in hme vecome wual tack
cGAN [175]:	
R(GT):	Andenmigt in time become social Kack
<b>Ours (S1):</b>	
R(GT):	whuch-might in time become social Lack
R(Generated):	xhud-migut in time Gecome social fact
Ours (S2):	
R(GT):	Aduchimight in time Vecome social Lact
R(Generated) :	rduch mgut in time become social fact

Figure 4.10: Results of fixing a degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by the CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images (better viewed in color), R (Generated): recognition by CRNN [134] trained on generated images (S2).



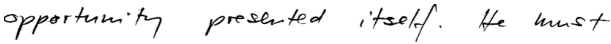
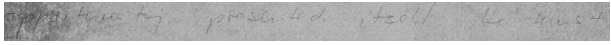
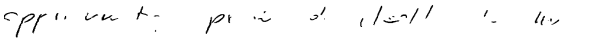
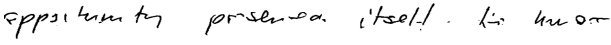
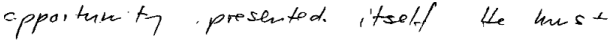
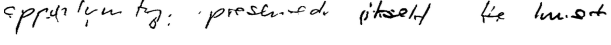
GT image:	
GT text:	opportunity presented itself . He must
R(GT):	oportunity presented itselyf . He must
Distorted:	
R(D):	o en a wao on
R(GT):	#A
Baseline:	
R(GT):	pr ". Un t. pri . lial! '- U.
cGAN [175]:	
R(GT):	rps hm to prsenca itsel! . Lis unor
<b>Ours (S1):</b>	
R(GT):	sportunty presented . itself He mese
R(Generated):	aportun-ty presented itself . Hhe must
Ours (S2):	
R(GT):	spen' in-ty . presen-ed- ritsel He lmist
R(Generated):	spantin- ty presened pitseld . be lmist .

Figure 4.11: Results of fixing a highly degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by the CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images, R (Generated): recognition by CRNN [134] trained on generated images (S2).


GT image:	<i>take charge, as it were, of the minds of the</i>
GT text:	take charge , as it were , of the minds of the
R(GT):	take change , as it were , of the misnds of the
Distorted:	
R(D):	Poe oroe wo h ore , of the onay of the
R(GT):	AHAH
Baseline:	<i>take charge as it were of the minds of the</i>
R(GT):	Ari charga na 4 eive of ikes nisndo of iln
cGAN [175]:	<i>take charge, as it were of the minds of the</i>
R(GT):	lode changa , a w iuse of this mondo of His
Ours (S1):	<i>take charge, as it were, of the minds of the</i>
R(GT):	Lotr chango , ao st wise , af tho minds g " the
R(Generated):	tokerchango , as it mese , of the minds of the
Ours (S2):	<i>take charge, as it were, of the minds of the</i>
R(GT):	Lat chanop , na d wore , of tho nundo of the
R(Generated):	lake chanop , hat wore , of the mundo of the

Figure 4.12: Results of fixing an extremely degraded handwritten line image. Errors made by the CRNN reading engine are shown in character level with the red color. R (GT): recognition by CRNN [134] trained on clean images, R (D): recognition by the CRNN [134] trained on degraded images, R (Generated): recognition by the CRNN [134] trained on generated images (S2).

## H-DIBCO Competitions

After demonstrating the suitability of our proposed method in recovering clean and readable images from highly degraded ones. In what follows, we validate it in H-DIBCO competition on handwritten document binarization, using H-DIBCO 2012 [148], H-DIBCO 2016, DIBCO 2017 [84] and H-DIBCO 2018 [146]. Since our model was designed to enhance line images with a size of  $128 \times 1024$ , we binarize H-DIBCO images in form of patches having the same dimensions. We compare with the recent state-of-the-art approaches, the winners of the different competitions [148] and the classic binarization methods [138, 164]. To clean H-DIBCO images, and since they are formed

of Latin script text, we used our pretrained model on the developed degraded-IAM dataset. Two scenarios were investigated: Using the model directly to clean the images, or fine-tuning it with a similar distribution before using it. For fine-tuning, we used the other DIBCO and H-DIBCO versions [146] and the Palm-Leaf dataset [26]. It is to note also that since the DIBCO datasets are not holding the text information, we removed the recognizer component during the fine-tuning process, we have frozen also the batch normalization layers of the generator and we trained the architecture for one only epoch to keep the learned knowledge of the degraded-IAM. During cleaning, we feed our model with the original degraded image in two forms: A normal condition and a vertically flipped version. Thus, we produce two instances of the recovered images. The flipped image is, then, re-flipped again to the normal condition. After that, a voting method is used to produce the final binarized image, by assigning zero to the pixel value (black) if it is indeed black in the two produced images by our model. We found that this led to a better result instead of using just one image condition.

We start our experiments with H-DIBCO 2012, the obtained results are given in Table 4.5. As it can be seen, our model leads to competitive results in the state-of-the-art approaches, with superiority in two metrics (PSNR and FM). However, we can see that the model proposed in [96] is better in terms of  $F_{ps}$  and DRD. Then, we tested our method on a more recent dataset which is H-DIBCO 2016 [83]. As presented in Table 4.6, our model gives the state of the art compared to all the methods in the three metrics PSNR, FM, DRD, and the overall average.

Next, we tested with DIBCO 2017, which contains a mix of handwritten and printed degraded document images. The results in Table 4.7 show that our model is not superior in this dataset, but it is competitive with the best approaches. We note that our model performance was affected by the type of binarized documents in this dataset, which contain several printed documents, while our model is designed essentially for handwritten text. Finally, we tested with the most recent H-DIBCO 2018 [146]. The results are shown in Table 4.8, where we compare with the most recent state-of-the-art results, the winner of the H-DIBCO 2018 competition, and the classic binarization methods. The performance of our model is superior to the different approaches in

Table 4.5: Comparative results of our proposed method on *H-DIBCO 2012* Dataset for document binarization. Avg = (PSNR + FM + Fps + (100 - DRD)) / 4.

Method	PSNR	FM	Fps	DRD	Avg
Otsu [138]	15.03	80.18	82.65	26.46	62.85
Sauvola et al. [164]	16.71	82.89	87.95	6.59	70.24
Guo et al. [86]	17.86	86.40	89.00	4.67	72.14
Zhao et al. [211]	21.91	94.96	96.15	1.55	77.86
Competition winner [148]	21.80	89.47	90.18	3.44	74.50
Kang et al. [96]	21.37	95.16	<b>96.44</b>	<b>1.13</b>	<b>77.96</b>
<b>Ours (S1)</b>	16.29	79.25	85.96	7.33	68.54
<b>Ours (S1) + Fine-tuning</b>	<b>22.00</b>	<b>95.18</b>	94.63	1.62	77.54

Table 4.6: Comparative results of our proposed method on *H-DIBCO 2016* Dataset for document binarization. Avg = (PSNR + FM + Fps + (100 – DRD)) / 4.

Method	PSNR	FM	Fps	DRD	Avg
Otsu [138]	17.80	86.61	88.67	7.46	71.40
Sauvola et al. [164]	16.42	82.52	86.85	5.56	70.05
Vo et al. [154]	19.01	90.10	93.57	3.58	74.77
Guo et al. [86]	18.42	88.51	90.46	4.13	73.31
He and Schomaker [75]	19.60	91.40	94.30	2.90	75.6
Zhao et al. [211]	19.64	91.66	<b>94.58</b>	2.82	75.76
Competition winner [83]	18.11	87.61	91.28	5.21	72.94
Bera et al. [160]	18.94	90.43	91.66	3.51	74.38
Kang et al. [96]	19.18	93.09	94.85	3.03	76.02
<b>Ours (S1)</b>	14.26	69.52	78.01	12.11	62.42
<b>Ours (S1) + Fine-tuning</b>	<b>21.85</b>	<b>94.95</b>	94.55	<b>1.56</b>	<b>77.44</b>

terms of PSNR, FM,  $F_{ps}$ , and average score.

Out of the obtained results in the different datasets, we can say that the classic thresholding methods [138, 164] have a moderate performance compared to the recent deep learning approaches. Also, we can notice that if our model uses only degraded-IAM for training, does not reach a satisfactory result because there is a domain gap between the training and testing data. However, fine-tuning our model with similar datasets leads to the best performance compared to all the state-of-the-art methods in H-DIBCO 2016 and H-DIBCO 2018. While having a competitive result with the best approach in H-DIBCO 2012 that is [96], where we obtain superior PSNR and FM scores. We can conclude also that our model is more suitable for binarizing the handwritten images since it was pre-trained on the developed degraded-IAM dataset before the fine-tuning stage.

Table 4.7: Comparative results of our proposed method on *DIBCO 2017* Dataset for document binarization. Avg = (PSNR + FM + Fps + (100 – DRD)) / 4.

Method	PSNR	FM	Fps	DRD	Avg
Otsu [138]	13.85	77.73	77.89	15.54	63.48
Sauvola et al. [164]	14.25	77.11	84.1	8.85	66.65
Zhao et al. [211]	17.83	90.73	92.58	3.58	74.39
Competition winner [84]	<b>18.28</b>	91.04	92.86	3.40	<b>74.69</b>
Kang et al. [96]	15.85	<b>91.57</b>	<b>93.55</b>	<b>2.92</b>	74.51
Bera et al. [160]	15.45	83.38	89.43	6.71	70.38
<b>Ours (S1)</b>	13.54	71.13	80.39	9.60	63.86
<b>Ours (S1) + Fine-tuning</b>	17.45	89.8	89.95	4.03	73.29

Finally, we show some qualitative results about the binarization performance in Figure 4.14 that demonstrate our method’s superiority compared to the other ones in this task. Also, we provide the binarization result of other images from the H-DIBCO

Table 4.8: Results for all methods on *H-DIBCO 2018* Dataset for handwritten document binarization. Avg = (PSNR + FM + Fps + (100 - DRD)) / 4.

Method	PSNR	FM	Fps	DRD	Avg
Otsu [138]	9.74	51.45	53.05	59.07	38.79
Sauvola et al. [164]	13.78	67.81	74.08	17.69	59.50
Adak et al. [146]	14.62	73.45	75.94	26.24	59.44
Soubgui et al. [174]	16.16	77.59	85.74	7.93	67.89
Tamrin et al. [181]	17.04	83.08	88.46	5.09	70.87
Zhao et al. [211]	18.37	87.73	90.6	4.58	73.03
Competition winner [146]	19.11	88.34	90.24	4.92	73.19
Akbari et al. [4]	19.17	89.05	93.65	4.80	74.26
Kang et al. [96]	19.39	89.71	91.62	<b>2.51</b>	74.55
Dang et al. [46]	19.81	91.26	93.97	3.42	75.40
Bera et al. [160]	15.31	76.84	83.58	9.58	66.53
<b>Ours (S1)</b>	13.88	65.06	73.46	12.86	59.89
<b>Ours (S1) + Fine-tuning</b>	<b>20.18</b>	<b>92.41</b>	<b>94.35</b>	2.60	<b>76.08</b>

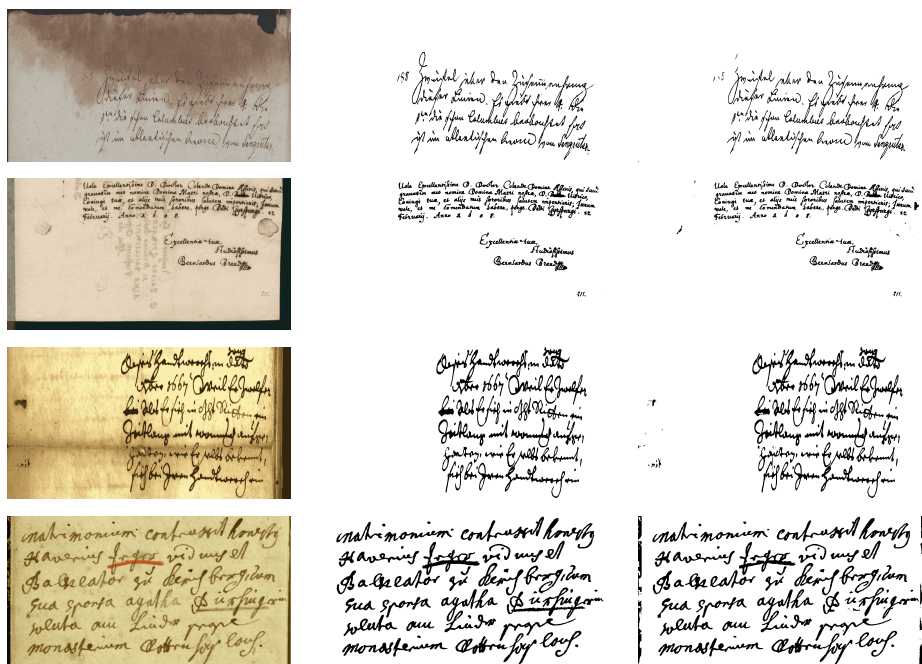


Figure 4.13: Results of our method in binarization of some samples from the *H-DIBCO 2018* dataset. Images in columns are: Left: original image, Middle: GT image, Right: Binarized image using our proposed method.

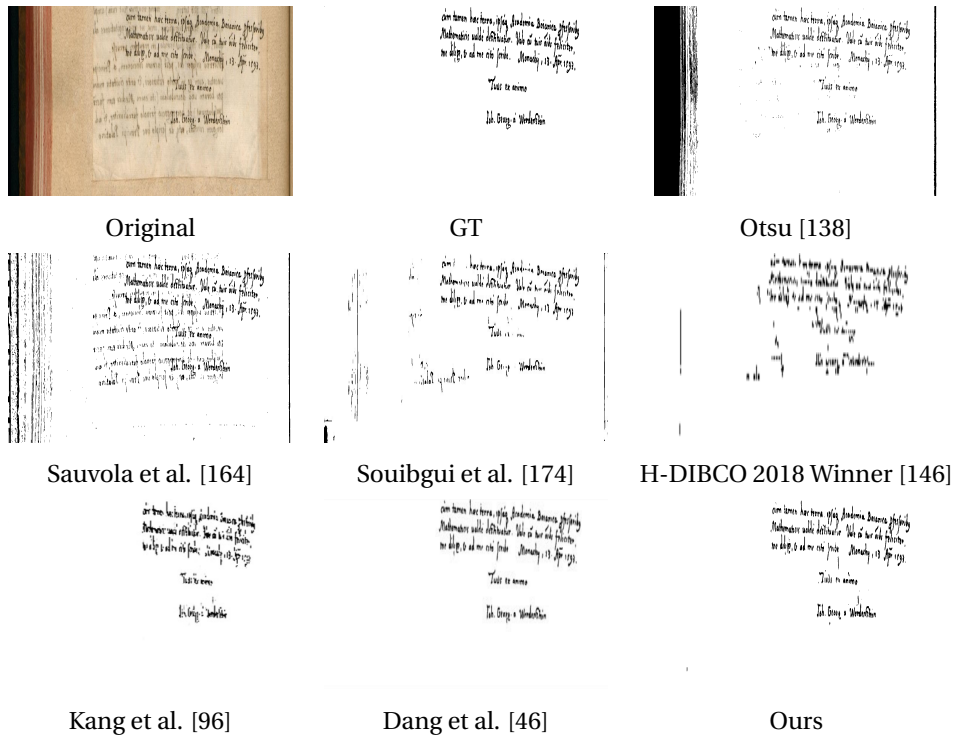


Figure 4.14: Results of the different enhancements on sample 4, from H-DIBCO 2018 Dataset.

2018 dataset in Figure 4.13 where we obtained images that are very close to the GT. Moreover, Figure 4.15 shows an example where our method can even complete the missing pixels (that do not exist in the GT image), to provide a more readable text.

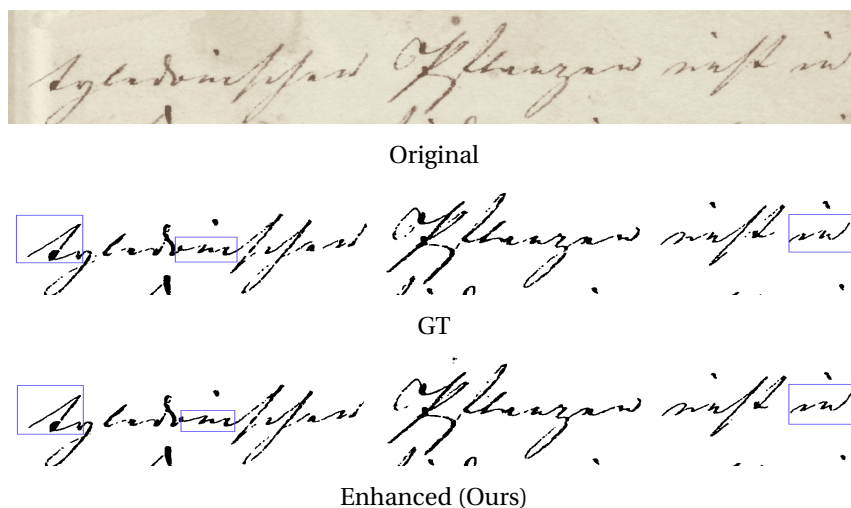


Figure 4.15: Qualitative results of our proposed method evaluated on a part taken from a sample from H-DIBCO 2018 Dataset (Pixels restoration).

### Dataset selection for the fine-tuning stage

Selecting the right dataset for fine-tuning will improve the binarization performance. Thus, in this section, we study the impact of the different datasets on the binarization process carried out on the H-DIBCO 2016. For all our experiments, we tested using the other variations of DIBCO and H-DIBCO (from 2009 to 2018), except DIBCO 2019 since it has different distributions in term of degradation and document types. Also, we tested including similar datasets which were developed for binarization task, namely, Palm-Leaf [26], Nabuco [156] and Bickley-diary [1]. Images contained in these datasets suffer from different kinds of degradation, such as water stains, ink bleed-through, and significant foreground text intensity. As it can be seen from Table 4.9, using our model trained only on degraded-IAM leads to poor results, thus, a fine-tuning stage is necessary. Using the H-DIBCO images for fine-tuning improves the performance with a slight superiority over using the DIBCO ones. This can be explained by the type of text because our model is pretrained to binarize the handwritten text. However, using H-DIBCO and DIBCO at the same time is a better option. Because DIBCO contains useful types of degradation that can be learned to be cleaned by our model even with printed text. Also, adding other datasets is sometimes useful, but at other times deteriorates the performance. This can be noticed when adding the Palm-Leaf dataset which improves the binarization while adding the Nabuco or the Bickley-diary is leading the model to learn non-suitable parameters for H-DIBCO 2016. This can be justified by the similar domain (degradation and text types) between H-DIBCO 2016 and Palm-Leaf distributions, while it is different from the other datasets.

Table 4.9: Impact of the fine-tuning data selection on the binarization performance on H-DIBCO 2016 Dataset.

H-DIBCO	DIBCO	Nabuco	Bickley-diary	Palm-Leaf	PNSR
					14.26
✓					18.25
	✓				18.08
✓	✓				18.10
✓	✓	✓			16.89
✓	✓		✓		17.98
✓	✓			✓	<b>21.85</b>
		✓	✓	✓	14.78
✓	✓	✓	✓	✓	17.63

## 4.4 Conclusion

In this Chapter, we proposed an architecture for handwritten document binarization based on GANs. Our method recovers the degraded images while conserving their readability by integrating an HTR to evaluate the enhanced image in addition to the discriminator. To the best of our knowledge, this is the first approach that includes textual information when performing the recovery process of handwritten documents. Experimental results proved the effectiveness of the proposed model in cleaning extremely degraded documents. We proved also that training an HTR model progressively on the images binarized by the generator at each iteration leads to a better performance in CER and WER. We used in this Chapter a CRNN as a recognizer, but it is worth noting that by using other HTR architectures we may obtain better recognition performances, since our method is flexible to integrate different ones. Moreover, we obtain the best performance compared to the state of the art in H-DIBCO benchmarks of degraded document binarization.





# Chapter 5

## An End-to-End Document Image Enhancement Transformer

---

*In this Chapter, we present a new encoder-decoder architecture based on vision transformers to enhance both machine-printed and handwritten document images, in an end-to-end fashion. The encoder operates directly on the pixel patches with their positional information without the use of any convolutional layers, while the decoder reconstructs a clean image from the encoded patches. Conducted experiments show a superiority of the proposed model compared to the state-of-the-art methods on several DIBCO benchmarks. Code and models are publicly available at <https://github.com/dali92002/DocEnTR>.*

---

### 5.1 Introduction

In recent times, Convolutional Neural Network (CNN)-based approaches have been widely applied to DIE related sub-tasks, like binarization [96, 91], deblurring [80], shadow [189] and watermark removal [174], etc. We show in the previous two chapters that the performance of these models has significantly improved over classical handcrafted techniques, however, such models do have their own set of drawbacks. Firstly, CNNs operate on regular grids, and using the same convolutional filter to restore different regions of a degraded document image may not be a sensible choice. Secondly, CNNs fail to capture high-level long-range dependencies as they are more suited for extracting low-level spatial information from images.

With the recent success of transformers in Natural Language Processing (NLP) [185,

50], its application to computer vision problems (like image recognition [54], object detection [28], visual question answering [22], handwritten text recognition (HTR) [159], etc.) also started getting more prominence. The self-attention mechanism proposed in [185] helps to capture global interactions between contextual features. Using local information combined with the knowledge of long-range global spatial arrangement is beneficial for an efficient image restoration model. This local information is often encoded in the patch content of an image and the large-scale organization is contained in the redundancy of this information across the patches of the image [47]. Contrary to CNNs, which process pixel arrays, Vision Transformers (ViTs) [54] split an image into fixed-size patches (eg. 8x8, 16x16, etc.), and they correctly embed each of them as latent representation and include positional embedding information as input to the transformer encoder. This allows encoding the relative location of the patches, along with both local (spatial) and global (semantic) long-range dependencies. The motivation for using ViTs for our overall proposed baseline model is that a missing/degraded patch in the distorted document image can be recovered from the neighboring patches' information with the power of the multi-head self-attention in ViTs, which quantifies pairwise global reasoning between them. Also, ViTs have been adapted in the overall model pipeline in an encoder-decoder-based setting, inspired by the concept of denoising autoencoders [186] used in the reconstruction of corrupted input data. The encoder is mapping the degraded image patches into latent representations, whereas the decoder is recovering a clean image version from those encoded representations.

The overall contributions of our work can be summarized in three folds:

- We introduce a simple and flexible Document image Enhancement Transformer (DocEnTr), an end-to-end image enhancement approach, that effectively restores and enhances a degraded document image provided as input. As far as we know, DocEnTr is the first pure transformer-based baseline that leverages the effectiveness of Vision Transformers (ViTs) in an encoder-decoder-based framework, without any dependency on CNNs.
- We have addressed document binarization as the key problem study in this work to investigate the power of DocEnTr architecture. Experimental evaluation shows that DocEnTr achieves state-of-the-art results on standard document binarization benchmarks (DIBCO), for both machine-printed and handwritten degraded document images.
- A comprehensive and intuitive case study has been dedicated in Section 5.4 to prove the utility of ViTs with its multi-headed self-attention mechanism in the task of document enhancement.

The rest of this Chapter is organized as follows. In Section 5.2 we review the state of the art. The Document Image Enhancement Transformer (DocEnTr) is described in Section 5.3. Section 5.4 contains an analysis of the extensive experimentation that has been conducted, including different quantitative and qualitative studies. Finally, in

Section 5.5 we draw the conclusions and propose open challenges for future research directions.

## 5.2 Related Work

### 5.2.1 Document Image Enhancement

The reader can refer to Section 2.2 for this part.

### 5.2.2 Transformers in Vision and Image Enhancement Tasks

In very recent years, transformers are behind the advances in deep learning applications. Transformer-based architectures first showed great success in NLP tasks [185, 50] for text translation and embedding, surpassing the previous LSTM approaches. This motivates many works to employ them for the vision tasks, for instance, classification [54], object detection [28], document understanding [200, 6, 117], etc. More related to this Chapter, transformers were also used for natural image restoration [119] and document images dewarping [57]. However, the architectures that were used in these later image and document enhancement approaches are still relying on the CNN feature extractors before passing to the transformers stage. Also, CNN is used to reconstruct the output image. Contrary, what we are proposing in this work is a full transformer approach that attends directly to the patches on the input images and reconstructs the pixels without the use of any CNN layer.

## 5.3 Method

The proposed model is a scalable auto-encoder that uses vision transformers in its encoder and decoder parts, as illustrated in Fig 5.1. The degraded image is first divided into patches before entering the encoder part. During encoding, the patches are mapped to a latent representation of tokens, where each token is associated with a degraded patch. Then, the tokens are passed to the decoder that outputs the enhanced version of patches. Unlike the CNN-based auto-encoders, which were usually employed for the document image enhancement tasks, the transformer auto-encoder is profiting from the self-attention mechanism which gives global information during every patch enhancement. Both decoder and especially encoder are inspired by the vision transformer (ViT) [54] architecture. We present more details of the model's architecture in what follows.

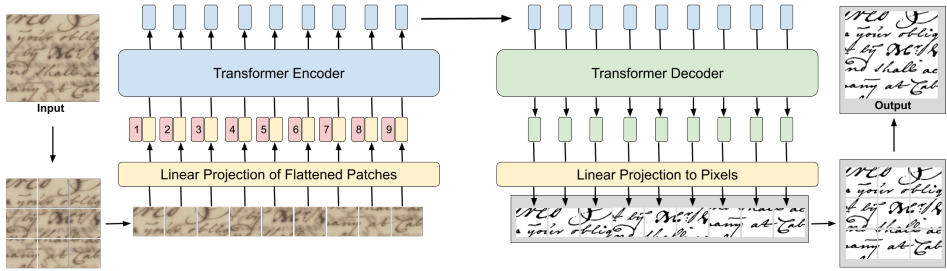


Figure 5.1: Proposed model: The input image is split into patches, which are linearly embedded, and the positional information is added to them (this is not shown in Figure because of space constraint). The resulting sequence of vectors is fed to a standard Transformer encoder to obtain the latent representations. These representations are fed to another Transformer representing the decoder to obtain the decoded vector, which is linearly projected to vectors of pixels representing the output image patches.

### 5.3.1 Encoder

In the encoding stage (left part of Fig.5.1), given an image, we divide it into a set of patches. Then, we embed these patches to obtain the tokens and add their positional information. After that, a number of transformer blocks are employed to map these tokens into the encoded latent representation. These blocks follow the same structure as [54], composed of alternating layers of multi-headed self-attention and multi-layered perceptron (MLP). Each of these blocks is preceded by a LayerNorm (LN) [10], and followed by a residual connection. The patch embedding size and the number of transformer blocks are set depending on the model size.

### 5.3.2 Decoder

The decoder part consists of a series of transformer blocks (having the same number as the encoder blocks) that take as an input the sequence of outputted tokens from the encoder. These tokens are propagated in the transformer decoder blocks and then projected with a linear layer to the desired pixel values. This makes each element of the output corresponding to a vector representing a flattened patch in the output image. The ground truth pixel values are obtained by dividing the ground truth (GT) clean image into patches (in the same way as the input degraded image) and flattening them into vectors. A mean squared error (MSE) loss is used between the model's output and the GT pixel patches to train the model.

### 5.3.3 Model Variants

Following a similar convention as previous works [50, 54], the proposed model configuration can be modified to produce different variants. In our experiments, we define three types of variants which are "Small", "Base" and "Large", as enlisted in Table 5.1. Evidently, setting a larger model requires more computational memory and training time since the number of model parameters is increasing. Thus, a trade-off between the model size and its enhancement performance must be taken into consideration.

Table 5.1: Details of our model variants

Model	Layers	Dim	Attention Heads	# Parameters
DocEnTr-Small	6	512	4	17M
DocEnTr-Base	12	768	8	68M
DocEnTr-Large	24	1024	16	255M

## 5.4 Experimental Validation

To validate our model, we use the datasets proposed in the different DIBCO and H-DIBCO contests [146] (except DIBCO 2019, because of the different degradation domain) for printed and handwritten degraded document images binarization and compare our results with the state of the art methods. Before these experiments, we conducted different investigations for a proper selection of the hyperparameters.

### 5.4.1 Choosing the Best Model Configuration

We begin our experiments by choosing the configuration that gives the best performance from our model variants (Small, Base, or Large). For training, each degraded image and its GT clean one is divided into overlapped patches with sizes  $256 \times 256 \times 3$ , the overlapping was set vertically and horizontally by half of the patches size (means 128). These resultant images (patches) will be used by our models as input and expected output (training data). For results evaluation, and same as the usual approaches [81], we utilize the following metrics: Peak signal-to-noise ratio (PSNR), F-Measure (FM), pseudo-F-measure ( $F_{ps}$ ) and Distance reciprocal distortion metric (DRD). We used in this experiment the DIBCO 2017 dataset, and the obtained results are given in Table 5.2. As it can be seen, a larger model gives a better result in all the metrics, but it requires more computation resources. Thus, we recommend using a Base model for a binarization task. Nevertheless, we will test as well the Large version in the following experiments.

Next, we do another experiment related to the input image size and the patches size that is used by our model. The reason behind this is that having different image sizes

Table 5.2: Results of varying the model size for the DIBCO 2017 dataset.  $\uparrow$ : The higher the better.  $\downarrow$ : The lower the better.

Model	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
DocEnTr-Small	18.29	91.06	93.82	2.78
DocEnTr-Base	18.69	91.66	94.11	2.63
DocEnTr-Large	<b>18.85</b>	<b>92.14</b>	<b>94.58</b>	<b>2.53</b>

and patch sizes can affect the binarization since the model is accessing to different types of information (from global to local). The obtained results using the Base model are given in Table 5.3. As it can be seen, slightly better performance is obtained using an input with a smaller size ( $256 \times 256 \times 3$  compared to  $512 \times 512 \times 3$ ). However, we can notice that the performance is highly improved when using a smaller patch size. The reason is that, by employing a smaller patch size, we make each patch of the image attend to more and much local patches during the self-attention. Thus, the model is looking to more and much fine information during the enhancement process with  $8 \times 8$  patch size. But, as before, using a smaller patch size means augmenting the model parameters, requiring more computation resources.

Table 5.3: Results of varying the input and patch sizes for the DIBCO 2017 dataset

Input Size	Patch Size	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
$256 \times 256 \times 3$	$8 \times 8$	<b>19.11</b>	<b>92.53</b>	<b>95.15</b>	<b>2.37</b>
$256 \times 256 \times 3$	$16 \times 16$	18.69	91.66	94.11	2.63
$256 \times 256 \times 3$	$32 \times 32$	17.57	89.37	91.99	3.44
$512 \times 512 \times 3$	$8 \times 8$	18.91	92.2	94.93	2.45
$512 \times 512 \times 3$	$16 \times 16$	18.66	92.15	93.89	2.54
$512 \times 512 \times 3$	$32 \times 32$	17.27	89.43	91.51	3.54

## 5.4.2 Quantitative Evaluation

After choosing the best hyper-parameters of the model, we conduct the experiments on the different datasets and compare our results with the related approaches. We begin by testing with the DIBCO 2011 dataset [82]. This dataset contains degraded document images with handwritten and printed text. For training, we use all the images from the other DIBCO and H-DIBCO datasets and the Palm Leaf dataset [26]. These images are split into overlapped images with size  $256 \times 256 \times 3$  before being fed to the model. The obtained results are given in Table 5.4, where we can notice a superiority of our method compared to the different variations of the related approaches. We choose to compare with different families of approaches: classic thresholding and

deep learning-based methods (whether based on CNN or cGAN). Our model DocEnTr-Base{8}, which means using the Base setting with a patch size of  $8 \times 8$ , gives the best PSNR and DRD compared to all the other methods. While the model DocEnTr-Large{16}, which means using the Large setting with a patch size of  $16 \times 16$ , leads to the second best performance in the metrics PSNR,  $F_{ps}$  and DRD. We note that for a computation reason, we were not able to train the Large setting with a patch size of  $8 \times 8$ .

Table 5.4: Comparative results of our proposed method on DIBCO 2011 Dataset. Thresh: Thresholding, Tr: Transformers.

Method	Model	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
Otsu [138]	Thres.	15.70	82.10	–	9.00
Savoula et al. [164]	Thres.	15.60	82.10	–	8.50
Vo et al. [154]	CNN	20.10	93.30	–	2.00
Kang et al [96]	CNN	19.90	<b>95.50</b>	–	1.80
Tensmeyer et al [183]	CNN	20.11	93.60	<b>97.70</b>	1.85
Zhao et al. [154]	cGAN	20.30	93.80	–	1.80
<b>DocEnTr-Base{8}</b>	Tr	<b>20.81</b>	94.37	96.15	<b>1.63</b>
<b>DocEnTr-Base{16}</b>	Tr	20.11	93.48	96.12	1.93
<b>DocEnTr-Large{16}</b>	Tr	20.62	94.24	96.71	1.69

After that, we test our model on the H-DIBCO 2012 dataset [148], which contains degraded handwritten document images while using the remaining datasets for training. As in the previous experiment, we use the other datasets for training with the same split size. The obtained results are shown in Table 5.5, where we can notice that our model gives the best performance in terms of PSNR and FM with the Base{8} configuration. We notice also that the other configuration gives competitive results compared to the other approaches.

Table 5.5: Comparative results of our proposed method on H-DIBCO 2012 Dataset. Thresh: Thresholding, Tr: Transformers.

Method	Model	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
Otsu [138]	Thres.	15.03	80.18	82.65	26.46
Savoula et al. [164]	Thres.	16.71	82.89	87.95	6.59
Kang et al [96]	CNN	21.37	95.16	96.44	<b>1.13</b>
Tensmeyer et al [183]	CNN	20.60	92.53	<b>96.67</b>	2.48
Zhao et al. [154]	cGAN	21.91	94.96	96.15	1.55
Jemni et al. [91]	cGAN	22.00	95.18	94.63	1.62
<b>DocEnTr-Base{8}</b>	Tr	<b>22.29</b>	<b>95.31</b>	96.29	1.60
<b>DocEnTr-Base{16}</b>	Tr	21.03	93.31	94.72	2.31
<b>DocEnTr-Large{16}</b>	Tr	22.04	95.09	96.00	1.64

Moreover, we tested with the more recent DIBCO 2017 dataset. In this dataset, our model achieves the best performance in all the evaluation metrics, as presented in Ta-



ble 5.6.

Table 5.6: Comparative results of our proposed method on DIBCO 2017 Dataset. Thresh: Thresholding, Tr: Transformers.

Method	Model	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
Otsu [138]	Thres.	13.85	77.73	77.89	15.54
Savoula et al. [164]	Thres.	14.25	77.11	84.1	8.85
Kang et al [96]	CNN	15.85	91.57	93.55	2.92
Competition top [84]	CNN	18.28	91.04	92.86	3.40
Zhao et al. [154]	cGAN	17.83	90.73	92.58	3.58
Jemni et al. [91]	cGAN	17.45	89.8	89.95	4.03
<b>DocEnTr-Base{8}</b>	Tr	<b>19.11</b>	<b>92.53</b>	<b>95.15</b>	<b>2.37</b>
<b>DocEnTr-Base{16}</b>	Tr	18.69	91.66	94.11	2.63
<b>DocEnTr-Large{16}</b>	Tr	18.85	92.14	94.58	2.53

Lastly, we test on the H-DIBCO 2018 dataset. Here, as shown in Table 5.7, the best performance is achieved by [91] based on cGAN. Anyway, we can notice that our model is still very competitive since it ranks second in the PSNR, FM, and  $F_{ps}$  metrics.

Table 5.7: Comparative results of our proposed method on DIBCO 2018 Dataset. Thresh: Thresholding, Tr: Transformers.

Method	Model	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}$ $\uparrow$	DRD $\downarrow$
Otsu [138]	Thres.	9.74	51.45	53.05	59.07
Savoula et al. [164]	Thres.	13.78	67.81	74.08	17.69
Kang et al [96]	CNN	19.39	89.71	91.62	<b>2.51</b>
Competition top [84]	CNN	19.11	88.34	90.24	4.92
Zhao et al. [154]	cGAN	18.37	87.73	90.60	4.58
Jemni et al. [91]	cGAN	<b>20.18</b>	<b>92.41</b>	<b>94.35</b>	2.60
<b>DocEnTr-Base{8}</b>	Tr	19.46	90.59	93.97	3.35
<b>DocEnTr-Base{16}</b>	Tr	19.33	89.97	93.5	3.68
<b>DocEnTr-Large{16}</b>	Tr	19.47	89.21	92.54	3.96

To summarize the quantitative evaluation, we demonstrate that our model gives good results compared to the state-of-the-art approaches. This was shown by obtaining the best results in most of the evaluation metrics with the H-DIBCO 2011, DIBCO 2012, and DIBCO 2017 benchmarks.

### 5.4.3 Qualitative Evaluation

After presenting the achieved quantitative results by our model, we present in this subsection some qualitative results. We begin by showing the enhancing performance of our method. This is illustrated in Fig. 5.2, where we compare our binarization results

with the GT clean images. As it can be seen, our model produces highly clean images, which are very close to the optimal GT images, reflecting the good quantitative performance that was obtained in the previous subsection.



Figure 5.2: Qualitative results of our proposed method in binarization of some samples from the DIBCO and H-DIBCO datasets. Images in columns are: Left: original image, Middle: GT image, Right: Binarized image using our proposed method.

Then, we present a quantitative comparison of our method with the related approaches. This is shown in Fig. 5.3, where we can notice the superiority of our model in recovering a highly degraded image over the classic thresholding [138, 164], CNN [96], and cGAN [91] methods.

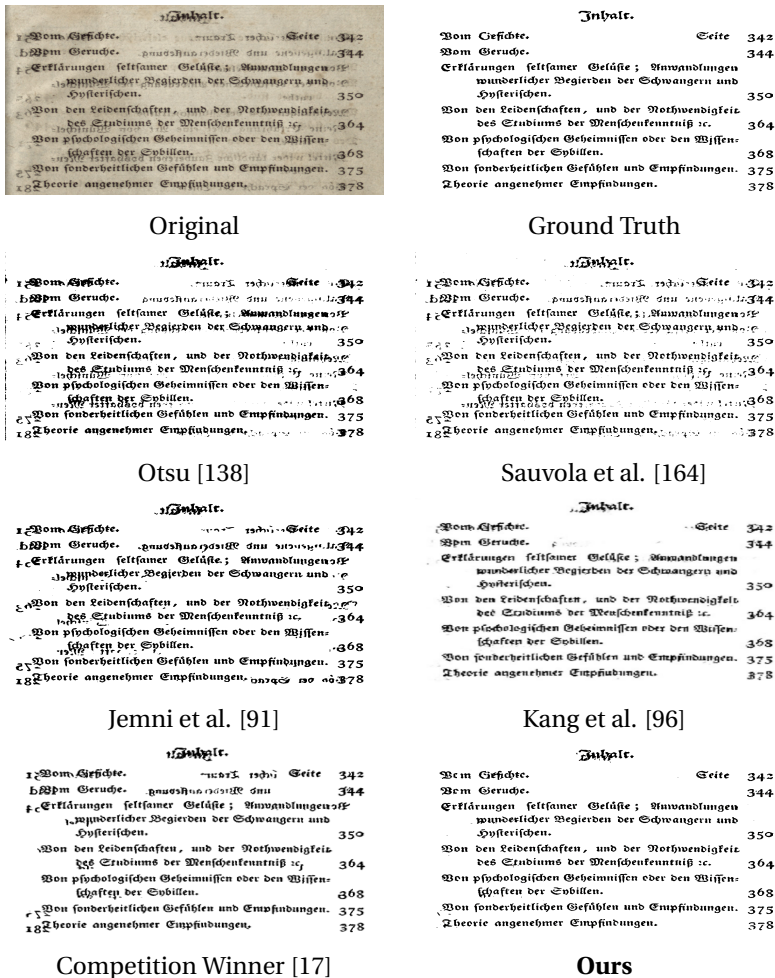


Figure 5.3: Qualitative results of the different binarization methods on the sample number 12 from DIBCO 2017 Dataset.

Degraded	Attention	Predicted	GT

Figure 5.4: Attention maps from the 2<sup>nd</sup> head of the last layer of DocEnTr{8} encoder. We display the self-attention for different (random) tokens.

#### 5.4.4 Self-attention Mechanism

As we stated above, our method differs from CNN-related ones by employing transformers to enhance the degraded document images. The self-attention mechanism used in the transformer blocks gives a global view to every token on the other tokens that represent the patches within the image for a better enhancing result. A visual illustration of the attention maps of the last layer from the encoder is given in Fig. 5.4. As can be seen, a token can attend to all the patches within the image. In these test cases, each token (patch representation) is focusing on the text elements, while ignoring the degraded patches. Thus, the attending patches are decoded later and projected to pixels while taking into consideration high-level global information from the attended neighboring patches that cover the full input image. We also notice that the attention maps are mostly matching the text of the GT images, which leads to a satisfactory binarization result that is closer to the GT. This supports the utility of using the transformers

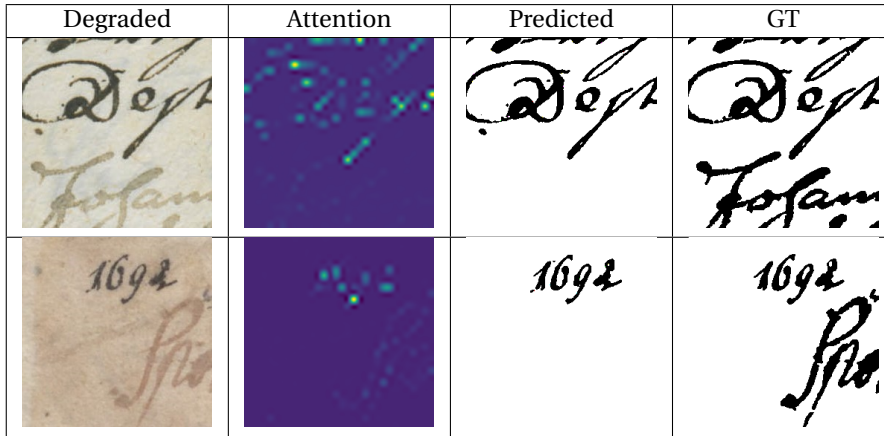


Figure 5.5: Attention maps from the 2<sup>nd</sup> head of the last layer of DocEnTr{8} encoder. We display the self-attention for different (random) tokens. (A failure case).

with their powerful self-attention mechanism in the image enhancement task. However, in other sample cases, as illustrated in Fig. 5.5, we observe that the attention maps are considering some portions of the text as a background region. Hence, the resultant enhanced image is removing foreground text because it considers it background noise. This explains the failure of the self-attention paradigm in these scenarios.

## 5.5 Conclusion

This Chapter presents a novel transformer-based architecture called DocEnTr for document image enhancement. To the best of our knowledge, this is the first pure transformer model addressing DIE-related problems. The model captures high-level global long-range dependencies using the self-attention mechanism for better performance. Quantitative and qualitative results on the DIBCO benchmarks prove the effectiveness of DocEnTr in recovering highly degraded document images. It is a simple and flexible framework that can also be easily applied to enhance other kinds of degradation occurring in document images (like blur, shadow, warps, stains, etc). These aspects will be investigated in future work. We also wish to investigate a self-supervised learning stage that can substantially benefit from large amounts of unlabeled data as well as test the utility of the proposed model in enhancing the OCR performances on the cleaned images.



## **Part II**

# **Document Image Recognition in Low Resource Data**





# Chapter 6

## Handwritten Text Recognition in Low Resource Data: State-Of-The-Art

---

*Recognizing low resource handwritten text images by a single machine learning model is challenging because: 1) the alphabet (or language) changes from one document to another, 2) there is a lack of annotated corpus for training and 3) touching symbols make the symbol segmentation difficult and complex. In this Chapter, we review the related work that address this problem in details.*

---

### 6.1 Introduction

Historical documents residing in archives and libraries contain valuable information of our past societies. Despite the mass digitization campaigns for preserving cultural heritage, many historical documents remain unexploited unless they are properly transcribed and indexed. One particularly interesting type of historical document is ciphered manuscripts.

Training data-hungry models based on deep learning in low-resource scenarios is challenging due to the scarcity of labeled data. This is particularly the case with modern Handwritten Text Recognition (HTR) systems when applied to manuscripts with rare scripts or unknown alphabets. For example, ancient civilizations used specific alphabets that are no longer used (e.g. cuneiform, Egyptian hieroglyphs) and historical ciphers (used in diplomatic and intelligence reports, secret societies, or private letters) often invented fanciful cipher alphabets to hide the content of the message [128].

Handwritten Text Recognition (HTR) systems are based on deep learning and require a significant amount of annotated data to reach a satisfactory performance. However, such systems suffer in low-resource scenarios. For example, data scarcity is a common problem when dealing with manuscripts with uncommon scripts or alphabets.

Historical ciphered manuscripts [128] is a typical case of low-resource handwritten text, where invented alphabets replace the known ones to encrypt the text and hide the content from undesired readers. Nowadays, many handwritten ciphered documents exist in archives consisting of military reports, diplomatic letters, records of secret societies, etc. Recognizing and extracting the hidden information is of great interest from the point of view of cultural heritage and history. However, manual transcription and cryptanalysis are costly both in terms of time and human resources. Therefore, the whole process needs automatic tools.

Because of the absence of context information in terms of language models and dictionaries, the automatic decryption of historical ciphered manuscripts is separated in two stages: transcription (HTR) and decipherment. The transcription step, which is the goal of this study, is a hard task due to the scarce annotated data to train, the paper degradation (typical in historical documents), and the changing alphabet across different ciphered manuscripts.

## 6.2 Related Work

### 6.2.1 Low Resource Manuscript Recognition: The case of ciphered text

A manuscript is considered a low resource when it contains rare symbols or unusual symbol sets. Thus, collecting a training set for this manuscript is difficult (especially a labeled one).

Nowadays, most of the developed approaches for HTR focus on natural known scripts (Latin [95], Arabic [72], Chinese [208], etc) and are based on deep learning architectures. Most models use Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), so they require a huge amount of annotated data and context information to learn in a supervised way the mapping function from the handwritten text image to the ground truth text class. These models are inappropriate for low-resource HTR for two main reasons. First, large annotated data are not available for training. Second, the alphabet of symbols usually changes from one manuscript document to another (especially in ciphered text), which makes the building of a single HTR model even more complex.

The research on the transcription of ciphered manuscripts is quite recent. In [60], an MultiDimensional Long Short-Term Memory (MDLSTM) [69] approach was proposed. The performance was satisfactory but at the cost of the time-consuming man-

ual data labeling. The method also required new, manual transcription for each new cipher. Instead, some unsupervised methods were introduced [14, 202] to avoid the costly human effort. Those approaches were segmenting the enciphered documents into isolated symbols and then clustering them. In those approaches, the enciphered document is first segmented into lines and isolated symbols, then a clustering algorithm is applied to group the visually similar symbols which then could be labeled with the target symbol used in the transcription. The main disadvantage of the clustering method turned is the segmentation of symbols because it was often inaccurate, provoking transcription errors. Similarly, researchers have opted for learning-free symbol spotting approaches [158, 25] for the transcription of ancient manuscripts (e.g. Egyptian hieroglyphs, cuneiform, or runes).

In summary, supervised methods obtain good performance but they require large amounts of labeled data, while unsupervised or learning-free methods can be applied when labeled data is not available but they lead to lower performance. Thus, to maintain high accuracy while reducing the human effort of manual labeling, few-shot learning is a promising alternative to use for hand-written text recognition [173]. A similar approach based on character matching was proposed in [205], although the experiments were mostly carried out on synthetic data, instead of on real historical or cursive manuscripts.

## 6.2.2 Handwritten Text Pseudo-Labeling

Pseudo-labeling models aim to take advantage of unlabeled data when training, which makes it a possible solution for low-resource manuscripts. In semi-supervised learning [215, 155], a few labeled data are used to start the process. For instance, in the label propagation approach based on distances [88, 193], labels are assigned from the unlabeled data (called pseudo-labels) to be used to reinforce the training. Similarly, in [110], the training started with some true labels that are gradually increased by pseudo labels. In [195] a shared backbone extracted features from the labeled, pseudo-labeled and unlabeled data at each iteration. Then, from the feature space, the reliable labels were estimated according to the distance with the true labels while the non-trusted labels were pushed away with an exclusive loss. Besides, a pseudo-labeling curriculum approach for domain adaptation used a density-based clustering algorithm in [40]. The idea was to annotate data with the same label set, but taken from a different domain.

In HTR, this strategy was hardly applied mainly due to the difficulties in character segmentation, since touching characters are common in cursive texts. In [61], labels were guessed at word level using keyword spotting. A confidence score was used to assign new labels to the retrieved words and enlarge the dataset. Furthermore, a text-to-image alignment was proposed in [113] following the mentioned strategy.

### 6.2.3 Data Augmentation and Generation for Handwritten Text

Data augmentation and generation are suitable solutions for deep learning models when data is limited. Classic data augmentation techniques used some image manipulation tricks [167], such as geometric transformations (e.g. rotations, resizing, warping), random erasing, color transformations, font thickness, flipping, etc. However, these methods need training data, and the augmented text is moderately realistic.

In the case of online handwriting, trajectory reconstruction approaches were introduced based on the kinematic theory of human movements [18, 144, 145] or by recurrent neural networks [70]. However, online information (e.g. stroke trajectory, speed, pressure) is not available in historical manuscripts, where only text images are available. Generative adversarial networks [68] and style transfer [67, 92] methods were utilized to generate the handwritten text from images. In [32], an approach to generate handwritten characters from an existing printed font was proposed. Also, in [94], cursive Latin words were generated conditioning on content (text) and a writing style. But these approaches need a huge set of annotated data to be trained on for each particular handwriting style, which is not available for low-resource applications. It is true that there are some attempts to use these techniques for few-shot by using only a few samples of each character class [15, 31, 44, 176]. Nonetheless, the results are still moderate in terms of quality and most of the methods are focusing on font translation while keeping the same text shape as the example that is conditioned on.

In this work, and to overcome the above limitations, we explore the use of BPL [106] to mimic the human ability to generate new unseen characters, while maintaining high quality and shape variation, from a single example.

### 6.2.4 Self-Supervised Learning

Due to extensive efforts on labeled data requirements of supervised models, this learning paradigm emerges as a way of exploiting the structured information contained in the data itself. Self-Supervised learning aims to obtain rich representations of an input modality by

designing pretext tasks that are used as auxiliary signals that are informative for a given downstream task. Initial approaches relied on auto-encoders [186] trained to remove artificially added noise from an image. Later, several approaches introduced other pretext tasks that provide rich signals to train a network as a feature extractor. Some pretext tasks employed were image colorization [206], jigsaw puzzle solving [136], patch ordering [51], rotation prediction [66] among others. Recent approaches rely on extensive image augmentation to maximize the agreement among paired samples and contrast with all possible negative samples [34, 35, 74, 204, 29, 30].

More recently, generative approaches like Masked Auto-encoders (MAE) [73] are introduced to predict a masked latent representation of patches. Similar ideas have been also explored in other recent works like BEiT [13] and PeCo [53] which adopt a discrete

variational autoencoder (VAE) to generate discrete visual tokens from the original image. Motivated by these works, we expand this generative learning framework to tackle text recognition and document enhancement tasks.



# Chapter 7

## A Progressive Few Shot Learning Approach for Low Resource Handwritten Text Recognition

---

*In this Chapter, we propose a few-shot learning-based handwriting recognition approach that significantly reduces the human annotation process, by requiring only a few images of each alphabet symbol. The method consists of detecting all the symbols of a given alphabet in a text line image and decoding the obtained similarity scores to the final sequence of transcribed symbols. Our model is first pretrained on synthetic line images generated from an alphabet, which could differ from the alphabet of the target domain. A second training step is then applied to reduce the gap between the source and the target data. Since this re-training would require annotation of thousands of handwritten symbols together with their bounding boxes, we propose to avoid such human effort through an unsupervised progressive learning approach that automatically assigns pseudo-labels to the unlabeled data. The evaluation on different datasets shows that our model can lead to competitive results with a significant reduction in human effort. The code will be publicly available in the following repository: <https://github.com/dali92002/HTRbyMatching>*

---



## 7.1 Introduction

Recognizing and extracting information from rare and historical manuscripts are important to the understanding of our cultural heritage, since it helps to shed new light on and (re-)interpret our history [129]. However, manual transcription is impractical due to the number of manuscripts, and automatic recognition is difficult due to the lack of labeled training data. Moreover, the problem becomes even harder in the case of ciphers because when the alphabet is invented, no dictionaries or language models are available to help in the training process.

Contrary to deep learning models, human beings are able to learn new concepts from one or a few samples only. Recent research has been conducted to imitate and simulate this ability. One of these recent approaches is called few-shot learning, requiring only a limited number of examples with supervised information [192]. In our previous work [173], we explored whether few-shot learning could be adapted to the recognition of various symbol sets in encrypted hand-written manuscripts.

Usually, HTR models must be trained on the particular alphabet to be recognized, and whenever the alphabet changes, the system must be retrained from scratch with samples from the new script. To avoid the cumbersome process of re-training, we treated the recognition as a symbol detection task: by providing one or a few samples of each symbol type in the alphabet, the system could locate the symbols in the manuscript. The model was generic and could be used for multiple scripts, while requiring only a small sample of labeled data on each new symbol type. The first experimental results obtained a good performance on encrypted manuscripts compared to the typical methods, while reducing the amount of labeled data for fine-tuning.

Nevertheless, the required labeled data in our few-shot model still implies a significant human effort: labeling a few pages with various types of symbols for fine tuning include manual transcription of thousands of symbols together with their corresponding bounding boxes. To alleviate this, we aim to minimize the time-consuming manual labeling effort by proposing an unsupervised learning approach that can automatically and progressively label the data by assigning *pseudo-labels* from the unlabeled hand-written text lines. Our method requires only a few shot of the desired alphabet: to perform the pseudo-labeling, the user crops a few samples – preferably 5 – of each symbol type thereby avoiding the annotation of text lines and the annotation of the bounding boxes. This means that the pseudo-labeled data is automatically obtained to fine-tune our model, with zero manual effort.

The main contributions of our work are: (i) We propose a few-shot learning model for transcribing hand-written manuscripts in low resource scenarios with minimal human effort. Our model only requires few, ideally five samples of each new symbol type, instead of annotating the entire text lines with the symbols and their bounding boxes. (ii) We propose an unsupervised, segmentation-free method to progressively obtain pseudo-labeled data, which can be applied to cursive texts with touching symbols. (iii) We propose a generic recognition and pseudo-labeling model that can be applied across different scripts. (iv) We demonstrate the effectiveness of our

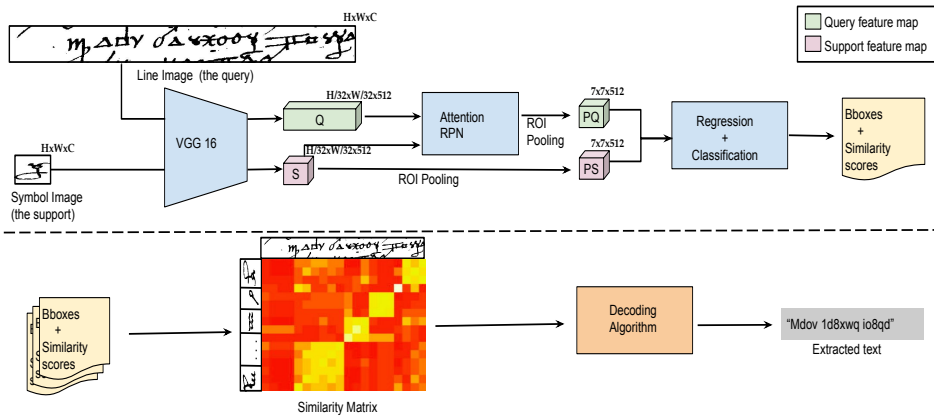


Figure 7.1: Our few-shot approach for handwriting recognition. Examples of each symbol in the alphabet are used as supports. Up: Detection of a support symbol in a handwritten line. Down: Construction of the similarity matrix from the predicted bounding boxes and its decoding to obtain the final text.

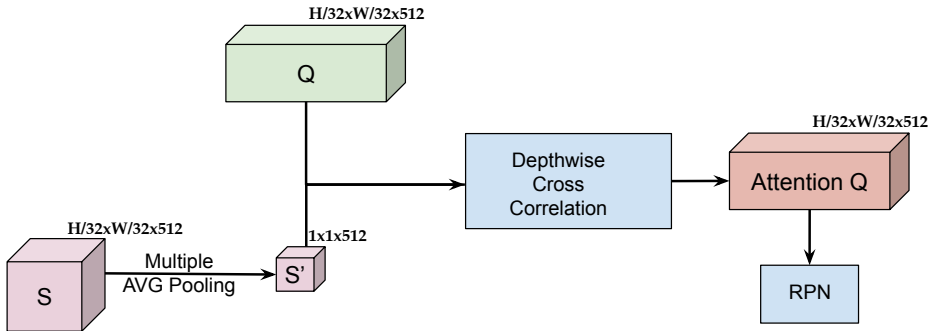


Figure 7.2: An illustration of the attention RPN: the support feature map is average pooled until obtaining a tensor with the shape of  $1 \times 1 \times 512$ . The obtained tensor is multiplied over depth with the Query feature map to obtain the attention Q, which is passed to the RPN for region proposing.

approach through extensive experimentation on different datasets with various alphabets, reaching a performance similar to the one obtained with manually labeled data.

## 7.2 Proposed Approach

In this Section, we describe our approach to handwritten text recognition by few-shot learning. First, our model is trained on synthetic data. We create text line images using various Omniglot symbol alphabets [106]. Then the model is fine-tuned using the pseudo-labeling approach with the specific alphabet of the target domain, in our case the hand-written manuscript. The involved steps are described in detail below.

### 7.2.1 Few-shot Manuscript Matching

As explained in Section 6.2, few-shot modeling for object detection has shown to be suitable for recognizing manuscripts in low-resource scenarios. In few-shot modeling, if the size of the alphabet is  $N$ , and we provide  $k$  examples from each symbol alphabet (named *shots* (or supports)), the task is considered as an  $N$ -way  $k$ -shot detection problem. In such setting, the model can be trained on certain alphabets with a sufficient amount of labeled data, and later, it can be tested on new alphabets (classes) with a few labeled examples only.

Our few-shot learning model, illustrated in Fig. 7.1, is segmentation free and works at the line level. As input, it takes the text line image with an associated alphabet in the form of isolated symbol images. In this step, from one to five samples of each alphabet symbol should be given. The two inputs (the line image as a query and a symbol image as support) are propagated in a shared backbone to derive two feature maps. Those are then used in the Region Proposal Network (RPN) with an attention mechanism to output proposals. The attention mechanism performs the depth-wise cross correlation between the support and query feature maps. As illustrated in Fig. 7.2, this is done by performing a multiple average pooling to the support feature map to obtain a shape of  $1 \times 1 \times Channels$  then multiplying it over depth with the query feature map. After the RPN stage, the Region of Interest (ROI) pooling is applied to the RPN proposals and the support image to provide two feature maps having the same size. These feature maps are representing the support image as well as the query regions that are candidates to match the support image. Those are combined together and passed to the final stage where the bounding boxes are produced with the class 1 (similar to the support) or 0 (different from the support symbol). For each labeled bounding box, a confidence score between 0 and 1 is predicted according to the similarity degree with the support image. We repeat this process for all supports (i.e. all the alphabet symbols) and take only the bounding boxes with a high confidence score (higher than a given threshold) to construct a similarity matrix between the symbol alphabet and the line image regions. This matrix serves as the input to the decoding algorithm, which provides the final transcription.

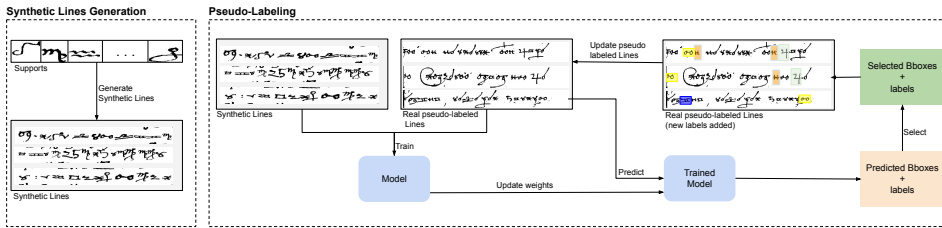


Figure 7.3: Our pseudo-labeling approach: In the beginning, synthetic lines are generated using the support set. Then, the pseudo-labeling phase starts. At starting, there is no pseudo-labeled data, so only synthetic lines will be used for retraining the model. Then, the model predicts symbols from the real unlabeled lines with the same script. The symbols with highest confidence score, namely pseudo-labels, are labeled and added with their predicted bounding boxes. Next, the model is retrained again using the synthetic lines and the pseudo-labeled symbols from real lines. The process is repeated until the full dataset is annotated.

## 7.2.2 Similarity Matrix Decoding

The decoding stage, detailed in Algorithm 1, takes the similarity matrix, traverses the columns from left to right, and decides for each pixel column the final transcribed symbol class among the candidate symbols. Concretely, for each time step, it chooses the symbol having the maximum similarity score. To minimize errors, a symbol is only transcribed if its bounding box is not overlapped by another symbol with a higher similarity value for a certain number of successive pixels. In our case, we used 15 pixels as a threshold. Despite its simplicity, this decoding method is effective for transcribing sequences of symbols. It can be considered also as a modified version of the Connectionist Temporal Classification (CTC) algorithm [71].

As mentioned before, our few-shot model is first trained on the Omniglot dataset: we synthetically construct lines to learn the matching in different alphabets. Then, at testing time, it can be used to recognize new symbols, requiring only a support set composed of a few examples of each new symbol class. However, in our previous work [173], experiments showed that the predictions can be significantly improved when we fine-tuned the model using some real text lines, due to the domain difference between the synthetic Omniglot symbols and the real historical symbols.

## 7.2.3 Progressive Pseudo-Labeling

Our proposed progressive data pseudo-labeling strategy consists in two stages described below.

**Algorithm 1** Similarity Matrix Decoding**Require:** $M$   $\triangleright$  Similarity matrix $rep\_thresh$   $\triangleright$  Repetition threshold**Ensure:**  $CharList$   $\triangleright$  Characters sequence $last\_max \leftarrow [-1, 0]$   $\triangleright [index, score]$  $repetition \leftarrow 0$  $maximums \leftarrow MaxInd(M)$   $\triangleright$  maximum index and score for each column $CanAdd \leftarrow False$ **for**  $maxi$  **in**  $maximums$  **do**  **if**  $maxi \neq last\_max$  **then**     $repetitions \leftarrow 0$      $CanAdd \leftarrow True$   **else**    **if**  $repetitions > rep\_thresh$  **and**  $CanAdd$  **then**       $CharList \leftarrow CharList \cup maxi[index]$        $CanAdd \leftarrow False$     **else**       $repetitions \leftarrow repetitions + 1$     **end if**  **end if****end for****Synthetic Data Generation**

Our few-shot model needs to be fine-tuned using data from the target domain (often with an unseen alphabet) to reduce the gap between the source and target domains. But since we aim to minimize the user effort, we restrain the demands on a support set of few samples from each new symbol alphabet. Hence, the user must only select up to 5 samples per symbol, called shots. From those shots, we automatically generate synthetic lines by randomly concatenating them in a line image. We tried to make those synthetic lines as realistic as possible. To do so, the space between characters was chosen randomly between 0 and 30 pixels. Also, before concatenation, we rotate each character randomly between -5 and 5 degrees. Moreover, we add some artifacts to the upper part and lower part of the line to simulate a realistic segmentation of a handwritten line. Those created lines compose our starting labeled set, since our model was only pre-trained on a different data domain, i.e. the synthetic Omniglot lines. This technique significantly improves the model prediction for unseen alphabets or scripts.

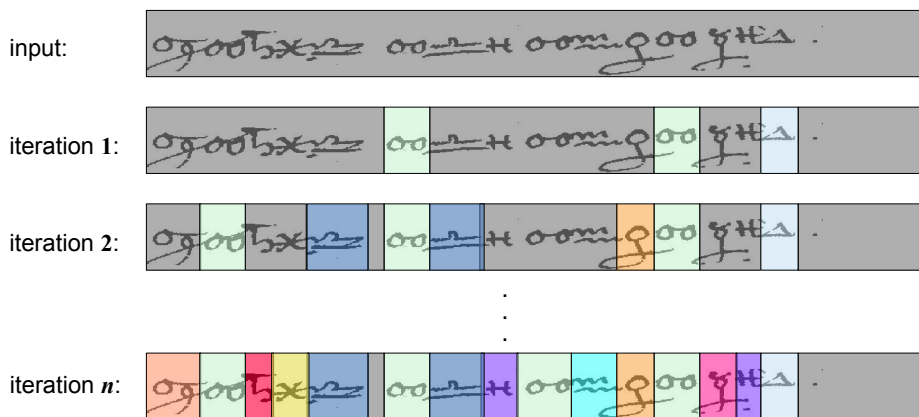


Figure 7.4: An example of pseudo-labeling of a line image. The background is colored in grey, while the predicted label classes at each time are shown in colors. Each symbol class is shown with a different color (best viewed in color).

### Pseudo-Labeling Process

After retraining our model with synthetic lines, we begin by annotating the unlabeled data. The process is illustrated in Fig.7.3. Of course, at the beginning, the pseudo-labeled set is empty (as no labels are available), so only the synthetic lines can be used for training. Then, we pass the real text lines through our model to get the predictions, which include the bounding boxes of the regions that are similar to the input alphabet images along with the assigned similarity score. Since a higher score means a more credible label, we choose the top-scored predictions as pseudo labels at this iteration. We experimentally found that the best option is to choose, at each iteration, 20 % of the training data size as the number of the new pseudo-labels. The obtained pseudo-labeled set will be joined to the synthetic set for the next training iteration. This process is repeated until the whole unlabeled set (i.e. all text lines) is annotated. In the case where it is not possible to add new pseudo-labels with a credible confidence score, we set a threshold of 0.4 as the minimum confidence score for assigning pseudo-labels. In fact, whenever the score is below this threshold, it is better not to label the symbol. Note that we label the handwritten lines without the need of segmenting them into isolated symbols. In this way, the remaining unlabeled symbols in the different lines at each iteration are considered as background during the next training. Fig. 7.4 shows an example of a handwritten line during the pseudo labeling process. At the beginning, the whole image is considered as a background. Then, the symbols with higher confidence scores are labeled in the first iteration, while the hardest ones will be labeled in the next iterations.

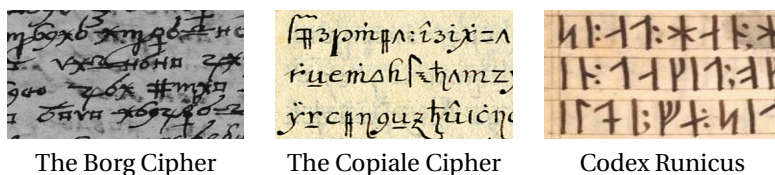


Figure 7.5: Examples of the three manuscripts with low resource annotated data.

## 7.3 Experiments

### 7.3.1 Datasets

For low-resource handwritten text, we chose two historical encrypted manuscripts and a manuscript with an old, no longer used alphabet, the Codex Runicus. Our choice of running experiments on encrypted manuscripts is motivated by the fact that ciphers contain a large variety of more or less fancy symbols instead of or in addition to using common alphabets and/or digits. In this work, we chose two encrypted manuscripts, namely the Borg and the Copiale ciphers, both containing a large variety of symbols. The Borg<sup>1</sup> cipher is a 408 pages long manuscript, originating from the 17th century. The entire manuscript is encoded with the exception of the first and last two pages, and some headings in Latin. The cipher consists of 34 different symbols, comprising from graphic signs to Latin letters and some diacritics. The Copiale cipher is a 105 page long encrypted manuscript from the mid-18th century. The cipher consists of 100 different symbols from Latin and Greek letters to digits along with a large number of graphic signs. The cipher has been transcribed and deciphered [102] and is freely available in high-resolution images<sup>2</sup>. The Codex Runicus<sup>3</sup> is a historical manuscript, the oldest preserved Nordic provincial law written on 100 parchment folios of 202 pages. Its symbol set consists of runes where each rune corresponds to a letter of the Latin alphabet. Fig. 7.5 shows examples of the two ciphers and the codex Runicus. As it can be seen, the Borg symbols are connected not only horizontally but also oftentimes vertically with many touching symbols making its recognition challenging. In the Copiale cipher, on the other hand, the symbols are clearly segmented but the size of the alphabet is large, which makes it a good challenge to test our approach on it. Similar to the Copiale cipher, the codex Runicus consists of clearly segmented symbols and a rare alphabet making it also a good case of low resource handwriting recognition. In our experiments, we exclude the symbols with low frequencies (that occur once or twice) in all manuscripts. We use 24 symbols from the Borg cipher, 78 symbols from the Copiale cipher, and 25 symbols from the Codex Runicus. Table 7.2 shows more information about our used datasets.

<sup>1</sup><https://cl.lingfil.uu.se/~bea/borg/>

<sup>2</sup><https://cl.lingfil.uu.se/~bea/copiale/>

<sup>3</sup><https://www.e-pages.dk/ku/579/>

### 7.3.2 Experimental Setup and Metrics

To carry out the experiments, we first trained our proposed few-shot handwriting recognition model using lines created from the Omniglot dataset only. Then, we retrained the model using synthetic lines created from the given 5 symbols (shots) as described above. This data is called Synthetic Data (SD). Afterward, we start predicting the labels and obtaining the Pseudo Labeled Data (PSD) by using the approach detailed in Subsection 7.2.3. We finally fine-tune the model with the pseudo-labeled data and compare its performance to the models that use Real Labeled Data (RLD) for training.

The model performance is measured by the Symbol Error Rate (SER) metric. It is the same as the Character Error Rate used in HTR. Formally,  $SER = \frac{S+D+I}{N}$ , where  $S$  is the number of substitutions,  $D$  of deletions,  $I$  of insertions and  $N$  is the ground-truth's length. Not surprisingly, the lower the value, the better performance.

We compare our approach with our previous few-shot model [173], the unsupervised [14, 202] and supervised [60] approaches for encrypted manuscript recognition.

## 7.4 Results

Table 7.1 shows the obtained results. The Borg manuscript is considered to be a hard case because of the overlapping symbols, which makes predicting correct bounding boxes challenging. Also, the writing style is variable. As it can be seen, using a few-shot method with real labels leads to a SER of 0.21, being considered the upper bound. But, this result is costly, since a user must manually annotate 1913 symbols, including their labels and bounding boxes. We also notice that the supervised MDLSTM with a larger training set, annotated at line level (but without any bounding boxes required), obtains a moderate result, probably due to the connected handwriting. We notice that the unsupervised methods are only useful when the segmentation of lines into isolated symbols is accurate, which is a costly and difficult task. Our few-shot model, trained on Omniglot only and tested on Borg, leads also to a poor result (an SER of 0.53) due to the big difference between the training and test domains. However, when using the pseudo-labeled data provided by our approach, we obtained an acceptable result of 0.24 SER, with a high gain in user effort because we only require 5 examples of each symbol, avoiding a time-consuming manual annotation.

The Copiale manuscript contains easy-to-segment symbols but with a larger alphabet size. As it can be seen from Table 7.1, the MDLSTM performs better on this dataset because of the larger labeled training lines and a unique handwriting style. However, our model achieves a competitive result by using less data than MDLSTM. Anyway, annotating these lines is costly, so a better choice is to automatically produce pseudo-labels. By using our pseudo-labeling process, we achieve a competitive performance, compared to the manually labeled data (a SER of 0.15 versus 0.11).

Finally, we test our method on the Runicus manuscript as an example of ancient



Table 7.1: Obtained Results on the different datasets. FT: Fine Tuning. Om: Omniglot. SD: Synthetic Data. RLD: Real Labeled Data. PLD: Pseudo Labeled Data. ULD: UnLabeled Data.

Dataset	Method	User Effort	Training → FT	SER
Borg	Unsupervised [202]	None	ULD	0.57
	Unsupervised [202]	Manual Segmentation	ULD	0.22
	Unsupervised [14]	Clusters Processing	ULD	0.54
	MDLSTM [60]	Manual Labeling	RLD	0.55
	Few-shot [173]	Manual Labeling	Om → RLD	0.21
	Few-shot [173]	5 shots	Om → NONE	0.53
	Ours	5 shots	Om → SD + PLD	0.24
Copiale	Unsupervised [202]	None	ULD	0.44
	Unsupervised [202]	Manual Segmentation	ULD	0.37
	Unsupervised [14]	Clusters Processing	ULD	0.20
	MDLSTM [60]	Manual Labeling	RLD	0.07
	Few-shot [173]	Manual Labeling	Om → RLD	0.11
	Few-shot [173]	5 shots	Om → NONE	0.39
	Ours	5 shots	Om → SD + PLD	0.15
Codex Runicus	Unsupervised [14]	Clusters Processing	ULD	0.06
	MDLSTM [60]	Manual Labeling	RLD	0.26
	Few-shot [173]	Manual Labeling	Om → RLD	0.05
	Few-shot [173]	5 shots	Om → NONE	0.40
	Ours	5 shots	Om → SD + PLD	0.09

document with a rare alphabet. This manuscript can be considered easier than ciphers because the symbol segmentation is easy and the alphabet size is moderate. Thus, an unsupervised clustering method can be also appropriate. Using our method with real labeled data, we obtain results that are better than without any fine tuning, with a SER of 0.05 and 0.40 respectively. When we compare the quality of our produced-pseudo labels against the manually created ones, we observe that, by using pseudo-labeling, we achieve a competitive result of 0.09 SER. This demonstrates the suitability of our method, because the performance is close to the one obtained with manual labels while significantly reducing the annotation effort.

We can conclude that our proposed pseudo-labeling method achieves good results when recognizing low resource handwritten texts, with an important decrease in the user effort for data annotation. The analysis of the human effort is detailed next.

### 7.4.1 Annotation Time Consumption

Manually annotating data is a time consuming task and it should be taken into account when using HTR models. Thus, in this section, we measure the time needed to

label the three datasets to illustrate the manual labeling effort. As shown in Table 7.2, the more lines and the bigger the alphabet size, the more time is required to label the symbols with their bounding boxes. For reference, we measured the required time for providing the shots for our method and compared it with the manual annotation time. We found that locating and cropping 5 examples of each symbol in the alphabet takes approximately 40 seconds. Thus the user needed to spend 16 minutes for Borg, 17 min for Runicus and 52 min for Copiale for providing the shots for all symbols in the manuscripts for our approach.

We can conclude that automatically providing pseudo-labels significantly minimizes manual effort with a minimal loss in recognition performance compared to the manual annotation.

Table 7.2: Required time (in minutes) for manually annotating the training lines.

Dataset	# Lines	# Symbols	# Classes	Time
Borg	117	1913	24	≈ 245
Copiale	176	7197	78	≈ 450
Runicus	56	1583	25	≈ 206

## 7.4.2 Pseudo-labeling Performance Analysis

Our proposed method progressively labels the dataset: we start by labeling easy symbols and progressively label the complicated ones. As a consequence, the accuracy of correctly labeling bounding boxes decreases as we select new pseudo labels at each iteration. We evaluate the quality of our pseudo-labeling approach on the three datasets by comparing the predicted bounding boxes and their corresponding pseudo-labels to the manually annotated ones. A predicted bounding box is defined as a correct detection if it has a minimum overlap (i.e. Intersection over Union: IoU) of 0.7 with the ground-truth box. We find that the more difficult the dataset is in terms of segmentation, alphabet size and similarity between symbols, the more the performance of our pseudo-labeling approach decreases and the more iterations in the labeling process are needed. For example, labeling accuracy for the Borg cipher was 74 % after obtaining all the labels. In Copiale, where symbols are easy to segment, the labeling accuracy reached 85 %. In Codex Runicus, we obtained the highest pseudo-labeling accuracy of 94 % because the symbol segmentation is easier than Borg and the number of classes is lower than in Copiale.

During our experiments, we found that it is better to continue the pseudo-labeling process despite the decreasing performance. The reason is that, although we might add some wrong labels, in general, the incorporation of difficult examples benefits the training and even a bounding box with a wrong label is still helping in the segmentation part. Moreover, the experiments show that there is a small difference in performance

between the manually annotated labels and our automatically produced ones, which encourages us to further improve our labeling process.

### 7.4.3 Selecting Threshold for Pseudo-Labeling

In our experiments we set a threshold of 0.4 before adding a character into the labeled set. This threshold is chosen after testing other values and finding that 0.4 is the optimal one. We show the results of the conducted experience in Table 7.3, where we tested different thresholds to select the pseudo-labels. The experiments were carried out on the Borg dataset.

Table 7.3: The symbol error rate when using different thresholds while pseudo-labeling the data. Thres.: Threshold

Thres.	SER	Thres.	SER	Thres.	SER	Thres.	SER
0.8	0.27	0.6	0.25	0.4	0.24	0.2	0.25

### 7.4.4 Is semi-supervised learning worth to use?

So far, we opt to use an unsupervised approach that starts from a few shots of the desired alphabet. However, the choice of starting by labeling some real lines (text and bounding boxes) and pseudo-labeling the rest is also a possible solution. We test this strategy on the Borg dataset as presented in Table 7.4. For comparison, we use two recent self supervised learning methods: masked autoencoders (MAE) [73] and UP-DETR [45]. Given that these methods were proposed for image classification and object detection, we adapt them to text recognition by adding a transformer decoder [185] in MAE and using our decoding algorithm in UP-DETR. As pre-training, MAE uses a set of Latin handwritten images because of the very few available Borg lines, while the UP-DETR is trained on the 117 unlabeled Borg lines. Then the fine-tuning is done using 20%, 30% and 50% of labeled Borg lines for both methods. The obtained results show that the more labeled lines, the better the performance. Despite the very few data that were used (the full training set is 117 lines), our method clearly outperforms the data-hungry MAE and UP-DETR, showing that they are not suitable for such low resource scenarios.

It is noteworthy that, in our method, if we start with more manually labeled lines, the amount of unlabeled lines to pseudo label is reduced, so the training time decreases. Overall, we can conclude that starting from only a few shots is a better solution with regards to the reduced manual effort, since the SER is slightly affected (we obtain 0.24 as SER using our unsupervised pseudo-labeling).

Table 7.4: Comparative results with self-supervised learning approaches in the semi-supervised scenario.

Method	Labeled lines	SER
MAE [73]	20 %	0.99
	30 %	0.99
	50 %	0.95
UP-DETR [45]	20 %	0.71
	30 %	0.70
	50 %	0.66
Ours	20 %	0.25
	30 %	0.24
	50 %	0.20

## 7.5 Conclusion

We have presented a novel pseudo-labeling transcription method for manuscripts with rare alphabets or few labeled data. Our method can significantly reduce the human labor of annotating historical manuscripts, while maintaining the recognition performance. The performed experiments on the enciphered and historical manuscripts confirmed the usefulness of our approach, with a significant reduction in user effort and a minimal loss in recognition performance.

Our few-shot model with pseudo-labeling is a significant extension of our previous work [173]. In fact, its simplicity makes it even applicable on top of other methods, like [205]. Also, for widely-used alphabets (like latin) but with few labeled data, pseudo-labels can be predicted to annotate the data and train usual fully supervised HTRs, which may lead to better results than the few-shot ones.

In future, we aim to enhance the quality of the provided labels to keep reducing the need of manual intervention. Also, we plan to extend our approach to cover more low resource datasets including other unknown scripts.



# Chapter 8

## A One-shot Learning Approach for Compositional Data Generation: Application to Low Resource Handwritten Text Recognition

### Abstract

---

*In this Chapter, we address the problem of low resource HTR through a data generation technique based on Bayesian Program Learning (BPL). Contrary to traditional generation approaches, which require a huge amount of annotated images, our method is able to generate human-like handwriting using only one sample of each symbol in the alphabet. After generating symbols, we create synthetic lines to train state-of-the-art HTR architectures in a segmentation-free fashion. Quantitative and qualitative analyses were carried out and confirm the effectiveness of the proposed method.*

---

### 8.1 Introduction

A typical solution to the lack of data is to create more examples for training via data augmentation or synthetic data generation. But these techniques [167] require training data. Moreover, and contrary to humans, deep learning models are known to fail on the

compositional nature of generation. Here by compositional, we refer to the generation of more complex items from simpler components/primitives, a process that humans can successfully do from just a single example [106]. Furthermore, generating data that covers the distribution of an alphabet using only a few examples of each symbol is a hard task for deep learning models.

In this work, we bridge between generation by compositionality and data scarcity by generating realistic samples to serve as ground truth to train HTR models. Concretely, our work is based on Bayesian Program Learning (BPL) [106], which uses simple programs to create more complex structures compositionally (i.e. to build rich concepts from simpler primitives/programs). However, BPL [106] was used to generate perfectly segmented symbols rather than sequences. This poses an important limitation for handwritten text recognition, because the text is a sequence of joined characters, especially in cursive handwriting. To overcome this limitation, we propose to create realistic text lines from the BPL-generated symbols. These symbols are generated starting with one single example of each symbol in the alphabet. As a result, the generated handwritten text lines can be used to train data-hungry HTR deep learning models for manuscripts with rare alphabets. As a study case, we use the Borg<sup>1</sup> cipher, a historical ciphered manuscript, written with an invented alphabet and containing many touching symbols, as shown in Figure 8.2. The goal is to transcribe such a difficult text with minimal user intervention (in terms of labeled data).

As far as we know, this is the first work that effectively uses BPL for Handwritten Text Recognition, as an example of the application of BPL for sequence recognition. The contributions of our work can be summarized as follows:

- We use BPL as a realistic symbol generation technique for handwriting recognition. The quantitative, and qualitative results and human studies demonstrate their effectiveness.
- We reduce the cost of annotation and human labor by automatically generating handwritten text lines by using just a single example per alphabet symbol.
- We experimentally show that the generated data benefits HTR models. Indeed, our approach outperforms the current state of the art in cipher recognition. This paves the way for it to be applied to other low-resource manuscripts.

The rest of the Chapter is organized as follows: BPL for handwriting generation is described in Section 8.2. Section 8.3 describes how BPL can be successfully used to generate text lines with high but realistic handwriting style variability. Section 8.4 analyzes the experimental results, whereas Section 8.5 presents the conclusions and future work.

---

<sup>1</sup><https://cl.lingfil.uu.se/~bea/borg/>

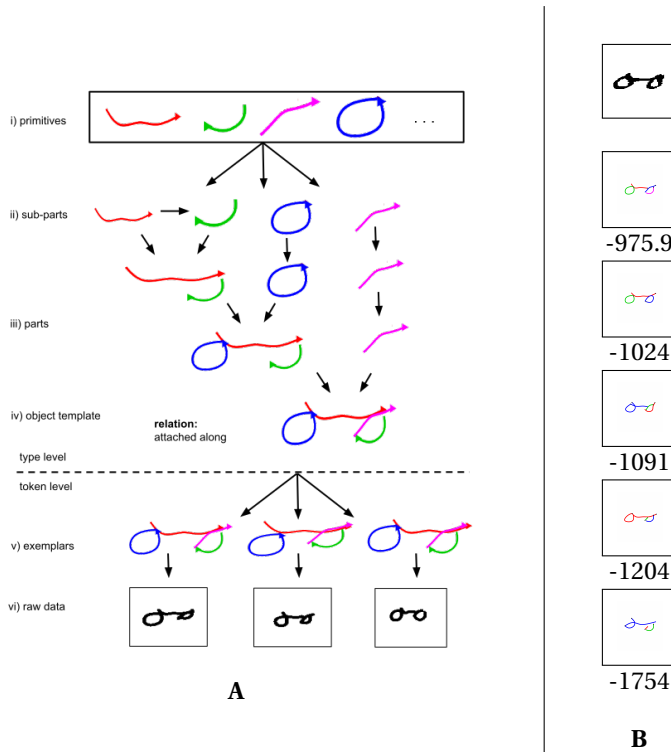


Figure 8.1: **(A)** A generative model of handwritten ciphered symbols. (i) From a library of color-coded primitives, new types are generated, (ii) combining these subparts, (iii) to further generating parts, (iv) and then defining simple programs by combining parts with relations. (v) Running these programs new tokens are generated, (vi) which are then rendered as raw images. **(B)** An image along with their log-probability scores for the five best programs. Parts are distinguished by color, with a colored flat back indicating the beginning of a stroke and a black arrowhead indicating the end.

## 8.2 Bayesian Program Learning (BPL)

Human beings have the ability to learn new concepts from a single example. Contrary, deep learning-based methods usually require tens or hundreds of examples to reach a human-level performance on recognition, generation, or parsing tasks. Thus, the generation of handwritten data from a few examples is still challenging. The Bayesian Program Learning (BPL) introduced in [106] showed a great ability to learn rich concepts compositionally and generate new examples from a *single unseen* concept, making it an ideal solution for the data scarcity problem.

As it can be appreciated from Fig. 8.1-A, BPL works in a hierarchical manner. At the highest layer, there are two levels called type level and token level (the dashed line in the middle). The type level consists of 4 steps which are sampling primitives, sampling



sub-parts, sampling sub-part sequence, and sampling relation. In BPL, primitives are defined as the smallest stroke in unit and time. More specifically, given a held-out set of data, BPL fits a Gaussian Mixture Model (GMM) on the normalized strokes according to their length and time information. Each of the centers of GMM cluster is treated as primitives and used as a starting point.

After obtaining the primitives, the number of primitives is sampled with  $P(\kappa)$ , where the distribution is obtained according to held-out data. Then, the number of sub-parts with  $P(n_i|\kappa)$  and sample sub-parts sequences with  $P(S_i|S_1, S_2, \dots, S(i_1))$  is sampled to decide which primitives should have relations, i.e. whether they should be combined into parts. Finally, BPL samples relations given sub-part sequence  $P(R_i|S_1, S_2, \dots, S(i_1))$  where there are four relations defined a priori for how the two strokes can be attached together. The two strokes can be attached “along”, “at start”, “at end”, or “independent”. All of the mentioned parts are combined with conditional probability to a program,  $P(\psi)$

$$P(\psi) = P(\kappa) \prod_{i=1}^{\kappa} P(S_i) P(R_i | S_1, \dots, S_{i-1}) \quad (8.1)$$

The token level parameters referred as  $\theta$ , consist of global re-scaling and a global translation of the center of mass of each sub-part sequence. Moreover, BPL adds variance to the created types in terms of start location, trajectory, and affine transformation so that the generated samples are as unique as possible. Token level parameters are distributed as  $P(\theta|\psi)$  and the end product of token level is  $P(I|\theta)$ .

At the inference phase, it produces a new image  $I^{(2)}$  given an image  $I^{(1)}$ . First, BPL reduces the line width of an image to one pixel and then runs a random walk algorithm to collect at most 10 parses of the  $I^{(1)}$ . An example of these parses can be found in Fig. 8.1-B. These parses are sorted according to the log probability of the random walk search and the most likely one is taken as the starting point. Afterward, a new image is generated according to the following formulation:

$$P(I^{(2)}, \theta^{(2)} | I^{(1)}) = \sum_{i=1}^K \sum_{j=1}^N \frac{w_i}{N} P(I^{(2)} | \theta^{(2)}) P(\theta^{(2)} | \psi^{[ij]}) \quad (8.2)$$

One of the main advantages of using BPL as a data augmentation is that it does not require huge training samples but more importantly, domain knowledge is minimized. For example, BPL can be trained on the Omniglot symbols, and later used on the Borg cipher symbols. Secondly, BPL can generate new exemplars from a *single unseen* example, whereas deep models are incapable of for the moment. Finally and most importantly, the output images have enough variability while keeping the main struc-

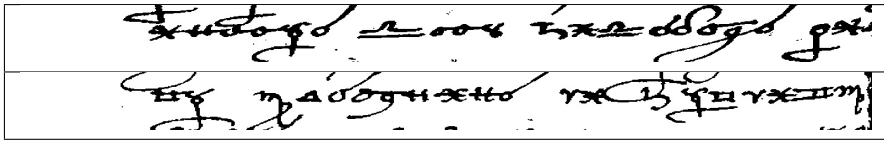


Figure 8.2: Two lines images from the Borg cipher. The image shows that there are frequent touching symbols in this manuscript, even between different lines.

ture to be used as a training set. In all of our experiments, we have used the code<sup>2</sup> to generate each symbol. The parsing includes a 'fast\_mode' option which skips the expensive procedure of fitting the strokes to the details of the image.

### 8.3 Handwritten Symbol Generation with BPL

In this section, we present the generation of cipher symbols using BPL. We quantify the effectiveness of the method and include a discussion of qualitative and human study results.

#### 8.3.1 Dataset

Borg is a 408 pages ciphered manuscript belonging to the 17th century. Its alphabet is composed of abstract, esoteric symbols, Roman letters, and some diacritics. Fig. 8.2 illustrates this handwritten text. As it can be seen, symbols are hard to segment, mainly because of the frequent symbol overlapping not only between consecutive symbols but also between the different lines. Following related works [14, 173], we have used 273 lines extracted from 16 pages for testing. Note that a pre-processing step (binarization and projections) has been applied to obtain those lines from the full pages of manuscripts. For data generation, we manually cropped 10 samples of each class in the alphabet.

#### 8.3.2 Data generation results - Symbol Level

In this section, we show the results of the BPL generation of Borg symbols and evaluate them according to a human study.

##### Qualitative Results

We provide two types of qualitative results. In Fig. 8.3-A, samples are generated using the top-right character. On the other hand, in Fig. 8.3-B, any example belonging to the same class but the top character can be used for generation. In other words, the

<sup>2</sup><https://github.com/brendenlake/BPL>

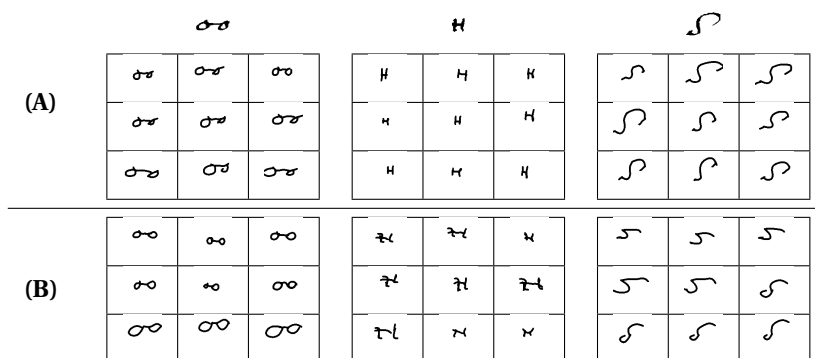


Figure 8.3: Generating new exemplars given one ciphered symbol. **(A)**: Conditioning on the same symbol (in-sample) shown on top of the nine-cipher grids. **(B)**: Conditioning on a different example of the same class (out-sample). The nine-character grids were generated by BPL.

generated examples of Fig. 8.3-B are not conditioned on the top character, rather it is conditioned on different examples from the same class of the top one. We will call the former in-sample and the latter out-of-sample examples.

As it can be seen for the in-sample generation from the Fig. 8.3-A, BPL mostly keeps intact the symbol structure while introducing variety. For example, from the first column, it can be seen that in some of the parts where it looks like “o”, BPL transforms it into a more open “o” (3rd row, 3rd column) and transforms the connection of lines. Moreover, it can change the line thickness and make the lines shorter or curved, see 3rd column of Fig. 8.3-A. These types of changes are compatible with human handwriting variability.

From the out-of-sample examples in Fig. 8.3-B, we can observe a much higher range of variations. The increase in variations is shown in both levels of type and token. At the token level, we can detect more diversity in affine transformations such as rotation (in the third column of the figure), scaling (third row of the figure), and translation in terms of the center. We also see a lot more diversity within symbols compared to the in-sample examples.

Apart from being realistic and introducing variety, what BPL offers cannot be obtained with other data generation techniques. Since BPL has used the actual human handwriting distribution, it is quite hard to reach a similar realism with any other ad-hoc generation technique.

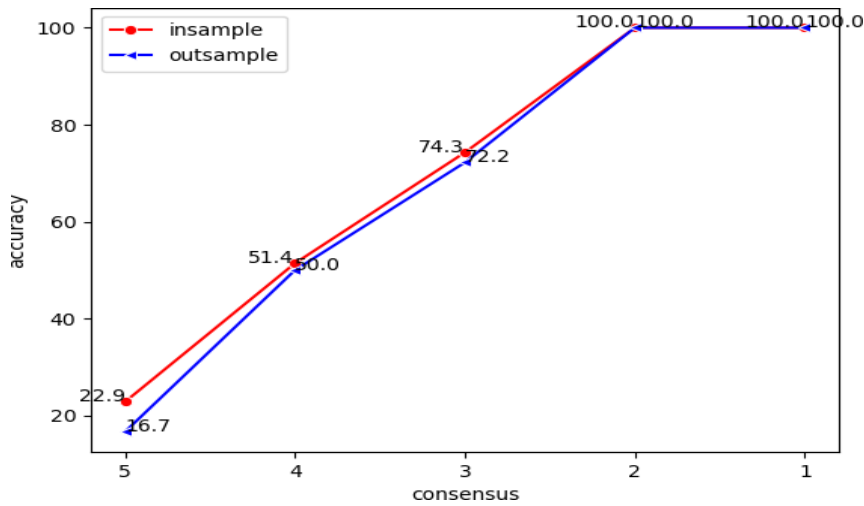


Figure 8.4: Result of the AMT human study where subjects are asked to match between real and generated images. The consensus seen in the  $x$ -axis represents the amount of agreement among subjects.

## Human Evaluation

Aside from providing qualitative results, we want to quantify the effectiveness of BPL in terms of how similar it is to its original examples. However, quantifying similarity in the handwritten text is difficult. For this reason, we have run a human study following a simplified version of our task formulation: Given an original symbol (query) and 5 options, human subjects have to choose the option that matches the query. In the experiment setup, 4 out of 5 options are BPL generated and a final option is “Not Sure”.

We set up 2 experiments to quantify the “realistic” generation: in-sample and out-of-sample generation. Both experiments follow the same procedure in which we provide Amazon Mechanical Turk (AMT) workers with a single symbol (a query) and ask them to find the most visually similar item to the query. In the first experiment, we pick the options generated from the query while for the second experiment options can be generated from any symbol but the query. The former experimentation will provide how accurate BPL is within in-sample distribution and the latter is how well it can match the out-of-sample distribution. For each class, we have selected 5 original symbols and 2 BPL-generated ones, giving us 10 tasks per symbol. Both experiments are set up using AMT, in which 5 workers had to answer each question. In total, for each experiment, we have 210 questions and 1050 answers from 5 different human subjects. The results of these experiments are shown in Fig. 8.4. We show the accuracy vs at least how many subjects chose the correct option. In other words, we plot what is the accuracy of at least  $n$  workers choosing the correct option. As it can be deduced from Fig. 8.4, from at least 5 workers to at least 3 workers, there is a steady increase. More-

over, it is quite remarkable to observe that an ad-hoc method that requires no training can still result in 22.9% or 16.7% accuracy for all workers correctly predicting.

However, we choose to focus on at least 3 workers correctly predicting because of the majority voting paradigm. Thus, according to at least 3 workers choosing the correct option, we get 74.3% and 72.2% accuracy for in-sample and out-of-sample. The first conclusion is that we have quantified our method's accuracy and it is reasonably high given that the probability of randomly selecting the correct option is 20%. The second conclusion is that there is not much difference between in-sample and out-of-sample accuracy, only 2.1%, which is encouraging considering the training procedure of our models. Finally, we can see that if we relax the assumption of majority voting, we are getting 100% accuracy for both experiments. This is quite promising since we have at least 2 human subjects that can match BPL samples to the query in all tasks in both types of experimentation.

## 8.4 Impact of Data Generation for HTR

In this Section, we describe the generation of text lines, the HTR methods, and the evaluation of cipher text recognition.

### 8.4.1 Data Generation results - Line Level

Most HTR methods recognize the text at the word or line level because it is hard to segment it into isolated characters, which is also the case with most cipher manuscripts. As a consequence, segmenting symbols and classifying symbols is not a feasible option. For this reason, we have created text lines to be used for training the HTR. Concretely, we take the symbols generated by BPL and horizontally concatenate them in a manner as much realistic as possible. We set the space between characters, chosen randomly between 0 and 30 pixels, and we rotate each character randomly between  $-5$  and  $5$  degrees. Also, we add some artifacts to the upper part and lower part of the line. The synthetic text lines created from the BPL-generated symbols are called the BPLL set.

For comparison, we also applied some data augmentation techniques (rotation, resizing, random thickness,...) on the real symbols and concatenated them to generate synthetic lines from the real symbols. We denote this set as DAL. Moreover, we created another set of lines by randomly mixing symbols from the two previous sets, resulting in three different sets of lines. A few samples from those lines are shown in Fig. 8.5. As can be seen, some noise was introduced to make the lines as similar as possible to the real ones. The training of the HTR models can be done using one of the created sets, and also by mixing sets. Thus, we used the following two forms of mixing:

- **Homogeneous Lines (HomL):** composed of lines from the BPLL set + lines cre-

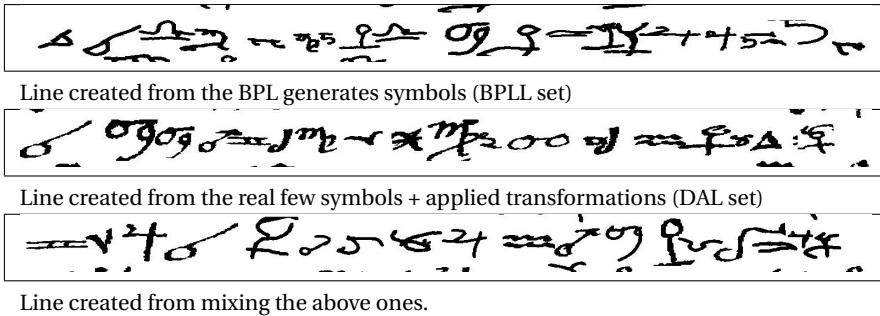


Figure 8.5: Examples of the three sets of lines, created by concatenating the symbols.

ated from the DAL set. In other words, we do not mix in the same line real symbols with BPL-generated symbols and vice versa.

- **Heterogeneous Lines (HetL):** composed of lines created from the mixed symbols (generated by BPL and data augmentation) + lines from the DAL set. In other words, we mix real symbols with BPL-generated symbols while generating a line and vice versa.

It is to note that we used three scenarios in this study: 10 samples, 5 samples, and 1 sample. That means that we start with only 10, 5, or 1 example(s) of each of the Borg symbols, respectively, to perform the data augmentation and the BPL generation in order to create the synthetic data from a low-resource alphabet.

### 8.4.2 HTR models and Evaluation Metric

After generating the synthetic lines, an HTR model can be trained for recognition. For this, we consider two options: A supervised model based on sequence to sequence with attention [95] and a few-shot learning-based model [173].

**Seq2Seq for HTR** The first method follows most of the HTR models, where the goal is to learn a mapping function from a line image  $X$  to a text  $Y$ . It is an attention-based sequence-to-sequence model, proposed in [95] and composed of three main parts: an encoder includes a CNN and a bi-directional Gated Recurrent Unit (GRU), an attention mechanism, and a decoder constituted from a one-directional GRU. Thus, given a line image as an input, the recognition is done character by character using the attention mechanism to produce the output text. This model showed competitive results in handwritten recognition with a huge amount of data using various Latin manuscript datasets for evaluation.

**Few-shot for HTR** Since we are using one or a few examples of each Borg symbol to generate the data (a few-shot generation), using a few-shot model for recognition could be suitable. Thus, we choose the approach proposed in [173], a segmentation

free (works at line level) method for historical ciphered handwritten text recognition. It consists in inputting a ciphered text line image with an associated alphabet as isolated symbols to a matching model, where one or a few examples (usually up to five) of each symbol should be given. Then, a similarity matrix between the line and the alphabet is outputted. After that, the recognized text is decoded from the matrix. Formally, if the size of the Borg alphabet is  $N$  and we provide  $k$  examples from each of the alphabet symbols for matching (the shots), the process is considered a  $N$ -way  $k$ -shot detection problem.

Table 8.1: Obtained results by different methods and settings: Real and synthetic data were tested with various sizes (# of ann. lines). # of generated samples indicates the number of images per each symbol, used to generate the synthetic lines.

Data Type	Model	#of ann. lines	# of gen. samples	$k$ -shot	SER
Real	Unsupervised [14]	None	–	–	0.54
	Few-shot [173]	None	–	5	0.53
	MDLSTM [60]	$\approx 81$	–	–	0.71
		$\approx 114$	–	–	0.66
		$\approx 148$	–	–	0.69
		$\approx 214$	–	–	0.55
Few-shot [173]	117	–	5	<b>0.21</b>	
Ours (HomL)	Few-shot [173]	1000	10	5	0.25
		1000	5	5	<b>0.25</b>
		1000	1	1	0.31
Ours (HetL)	Few-shot [173]	1000	10	5	0.30
		1000	5	5	0.28
		1000	1	1	0.41
Ours (HomL)	Seq2Seq + Attention [95]	1000	10	-	0.70
		1000	5	-	0.69
		1000	1	-	0.77
Ours (HomL) + Real	Few-shot [173]	117 + 117	5	5	<b>0.20</b>
Ours (HomL)	Seq2Seq + Attention [95]	2500	5	-	0.50
		5000	5	-	0.48
		10000	5	-	0.47
		20000	5	-	0.47

We have chosen this model because it has been applied to ciphers in a few shot scenario. The method is trained on synthetic alphabets (e.g. Omniglot [106] constructed lines) and tested on real ciphered data, requiring only the support set. However, the results show that this model can obtain better results when it is fine-tuned on some real data.

**Evaluation Metric** The evaluation of the ciphered text transcription is done according to the Symbol Error Rate (SER) metric. It is similar to the Character Error Rate (CER) for text recognition. Formally,  $SER = \frac{S+D+I}{N}$ , where  $S$ ,  $D$  and  $I$  are the numbers of required substitutions, deletions, and insertions, respectively, while  $N$  is representing

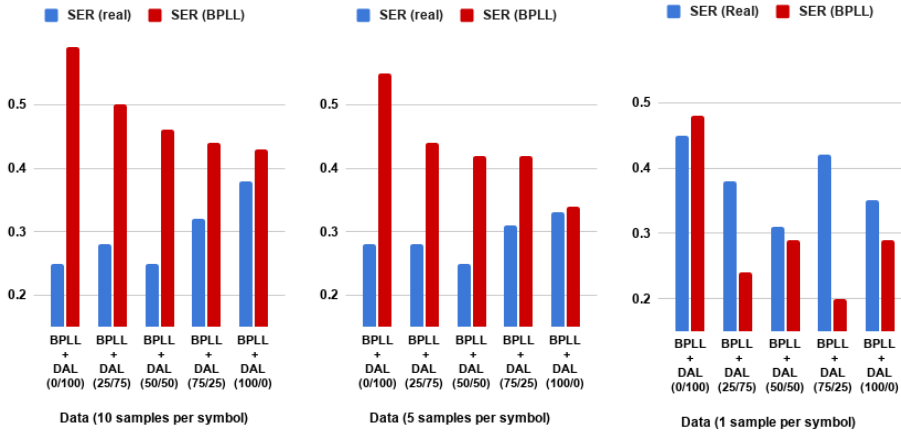


Figure 8.6: SER of testing with real Borg lines and synthetic BPLL lines, using different mixing settings and conditioning on different numbers of samples for generation.

the ground-truth's line length in term of symbols.

### 8.4.3 HTR Results

We begin by finding the best setting to mix the created datasets, then we compare the best performance with the state-of-the-art methods for transcribing the Borg ciphered manuscript using real data.

**Effect of mixing:** We took lines from BPLL and DAL sets to find the best mixing setting for the HomL mixed set, defined above. The assumption is that using this mixing we can obtain better results than using separate sets. Because, as it can be seen from Fig. 8.5, BPLL lines are visually rich: the writing style variation of each symbol is realistic, while DAL lines are visually similar to the real Borg lines. To find the right amount of lines from each set that should be added to the total mix, we perform an experiment of varying the percentages of the BPLL and the DAL, and calculating the SER at each time. The used model for this training is the one presented in [173] which shows good results for the Borg manuscript. As shown in Fig. 8.6, the performance of the model trained using only the BPLL is not optimal, but it is stable using different amounts of shots [0.33, 0.38]. On the other hand, by using only the DAL based on classic data augmentation techniques, the performance decreases if we reduce the number of examples per each Borg symbol that are used to create the data (from 0.25 using 10 samples to 0.45 using 1 sample). Mixing both sets, leads to better performance, especially if the mixed data is composed of 50 % from each set. Thus, we can conclude that adding the same number of DAL to the BPLL lines acts like a regularization technique. That is why we will keep this setting in the next experiments to compare our generated data with using real Borg data to train models.



**SOTA results:** The obtained results using different approaches and data settings are presented in Table 10.3. We compare the performance when using real Borg lines (where we can apply unsupervised or supervised learning), versus using our synthetically generated lines. As expected, annotated data-free approaches, such as the unsupervised [14] or few-shot without fine-tuning [173] (although providing one or few examples from each symbol to be used as supports is still needed) obtains very poor results. The reasons are the high degree of similarity among the Borg symbols and the difficult symbol segmentation in the unsupervised approach, whereas in the few-shot method the problem is the difference of distribution between the Borg dataset and the Omniglot one that was used to train. It is better, hence, to train the segmentation-free models with annotated data. This leads to two options: Using real annotated data or using our synthetically created one.

In the case of having a high amount of annotated lines, training an MDLSTM in a supervised way could lead to a good result. But, in our experiments, since the maximum number of available annotated lines is 214, and knowing that the Borg manuscript has different handwriting styles, the results are still moderate. Note that, of course, the performance improves when providing more annotated pages, which require more user effort. With the few-shot method, however, a much better result of 0.21 SER is obtained with few annotated data (117 lines for fine-tuning the model pretrained on Omniglot). But, even the annotation of these 117 lines at the symbol level (i.e. providing the bounding box of each one) is a time-consuming task. Note that when performing the human annotation experiment, we found that those lines require approximately 4 hours to be labeled.

To reduce this effort, the same model is fine-tuned with our synthetically created lines. Using the HomL set, the results are slightly diminished, from 0.21 to 0.25 as SER. But, we believe that this difference of 0.04 is not worth it because it implies annotating 117 lines. Instead, by using our BPL-based approach, the user just needs to provide 5 examples of each Borg symbol. Moreover, we can obtain a SER of 0.31 which is also competitive, when using only 1 example per symbol, to generate the lines. This proves the effectiveness of our synthetically generated data in replacing the real one, with a huge gain in annotation effort, and a minimal decrease in recognition performance. We also notice that using 10 or 5 examples to generate data gives the same results when testing instead of improving. This can be explained by increasing the out-of-the-sample matching, which may require using more data to cover it. We note also that using the other set (HetL) for training leads to a minor performance.

For comparison sake, we tested the Seq2Seq [95] model, with the same lines. But results were unsatisfactory because it needs much more data to be trained than MDLSTM. Hence, we generated thousands of text lines for training using the 5 shots setting. However, we can see that the performance stabilizes after using a certain amount of lines. The reason is that the generated sample variation is quite limited since we are only using 5 samples. Thus, we can conclude that generating fewer samples and training the few-shot model is a much better option.

Finally, we can see that when mixing the real Borg lines with the same amount

of lines generated synthetically from 5 examples by BPL, we obtain the best result, concretely 0.20 SER, which indeed outperforms the state of the art of recognizing this manuscript.

#### 8.4.4 Latin Handwritten Text

Next, we select a modern manuscript to further investigate the applicability of BPL generation. For this purpose, we take the English manuscript from the IAM dataset and simulate the low resource scenario by taking only 73 lines belonging to the writer 552. Then, we cropped 1 example from each English character (upper and lower cases) to generate data using our method. After that, we train the few-shot model in a 1 shot setting (note that we are not using any labeled text lines, we are just using 1 labeled example from each isolated character). The obtained results are shown in Table 8.2. As it can be seen, applying our model without the BPL generation leads to 0.35 as CER. While by adding the BPLL lines, we boost the performance to 0.31 as CER, which demonstrates the utility of our method.

Data Type	Model	CER
DAL	Few-shot [173]	0.35
HomL	Few-shot [173]	0.31

Table 8.2: The results on IAM dataset, simulating the low resource handwritten recognition. The numbers are in terms of character error rate (lower is better).

## 8.5 Conclusion

In this Chapter, we have used a one-shot approach for compositional handwritten text generation and we have demonstrated its effectiveness for low resource text recognition, as an example of sequence recognition. Although we have taken historical ciphered manuscript recognition as a study case, it can be applied on any other alphabet or script.

Our method uses BPL to generate synthetic symbols from a few real examples. Afterward, synthetic lines were created to train machine learning algorithms for HTR. From the experiments, we can say that the created data leads to competitive results compared to using a real annotated dataset, with a significantly reduced manual annotation effort by a huge margin. Moreover, we have achieved the state of the art in Borg ciphered text recognition when combining it with real data for training.

As future work, we will investigate more realistic approaches to creating text lines from the generated symbols, for instance, the impaired domain translation methods. Moreover, we will investigate using the compositional generation to directly create words or lines instead of isolated symbols, with the possibility of applying this to different manuscripts.



# Chapter 9

## A Self-Supervised Transformer Autoencoder for Text Recognition and Document Enhancement

---

*In this Chapter, we propose a Text-Degradation Invariant Auto Encoder (Text-DIAE), a self-supervised model designed to tackle two tasks, text recognition (handwritten or scene-text) and document image enhancement. We start by employing a transformer-based architecture that incorporates three pretext tasks as learning objectives to be optimized during pretraining without the usage of labeled data. Each of the pretext objectives is specifically tailored for the final downstream tasks. We conduct several ablation experiments that confirm the design choice of the selected pretext tasks. Importantly, the proposed model does not exhibit limitations of previous state-of-the-art methods based on contrastive losses, while at the same time requiring substantially fewer data samples to converge. Finally, we demonstrate that our method surpasses the state-of-the-art in existing supervised and self-supervised settings in handwritten and scene text recognition and document image enhancement. Our code and trained models will be made publicly available at <https://github.com/dali92002/SSL-OCR>.*

---

### 9.1 Introduction

In recent times, self-supervised learning paradigms have gained a lot of attention due to their ability to benefit from massive unlabelled data which is easily accessible from

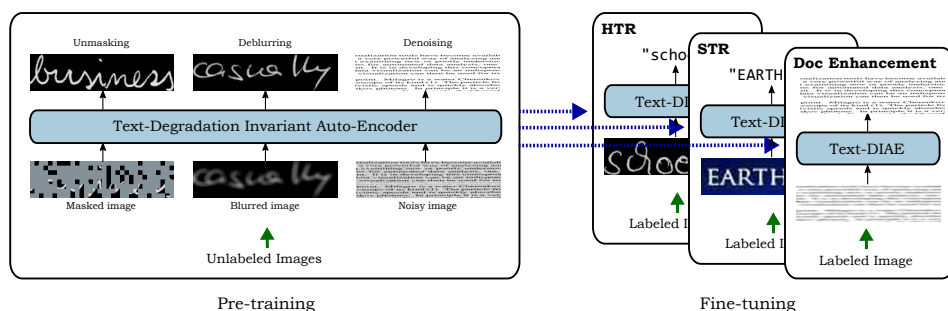


Figure 9.1: **Text-Degradation Invariant Auto-Encoder (Text-DIAE)**, we employ image reconstruction pretext tasks at pretraining. Masking, blurring and adding noise are employed to learn richer representations.

different sources. However, applying these approaches remain quite limited in the domains of optical character recognition (OCR), handwritten text recognition (HTR), and document image enhancement, which motivate us to tackle the problem in this study.

Common computer vision pipelines using self-supervised frameworks employ a pretext-task (e.g. relative position prediction of patches [51], contrastive views [34], image inpainting [140], etc.) to learn visual representations for solving down-stream tasks like classification, object detection and so on. Current self-supervised paradigms [30, 29, 34, 37] have adapted transformers [185] to learn visual representations from unlabelled images which are semantically meaningful. More recently, generative self-supervised approaches [73, 13, 53] using auto-encoders have been used to learn representations in the feature space through image patches and visual tokens.

Closely related to our work, some contributions in visual representation learning were addressing text recognition (HTR) [2, 21, 120] and Scene-Text Recognition (STR) [2, 207]) and image enhancement [118]. Despite the performance gains, there are some drawbacks of such models: (1) independent sequences of tokens are treated as single data points, which can cause misalignment of similar sequences among a batch, (2) considerable batch size requirements to define negative contrastive pairs, (3) considerably slow convergence rates.

For humans, reading text in noisy scenarios is possible because of the ability or reconstructing the degraded regions and predicting the missing/blurry content [78, 48]. Incorporating such an ability in a model could immensely help in the restoration, recognition, and understanding of characters and symbols, considering that text carries rich linguistic information that allows humans to reason explicitly according to context. In order to endow this human-specific skill to our models, we present in this Chapter a new self-supervised framework called Text-Degradation Invariant Auto-Encoders (Text-DIAE) inspired by the principle of denoising autoencoders [186], as depicted in Figure 9.1. Our model focuses on exploring the dynamics of learning representations under different degradation scenarios. Specifically, we propose the usage

of a robust self-supervised auto-encoder along with customized pretext tasks (masking, blur and background noise) that are designed to specifically tackle two different downstream tasks: text recognition (HTR and STR) and document image enhancement (document binarization, document deblurring). As a consequence, the choice of the proxy tasks has been realized to learn useful representations for solving these specific downstream tasks using unlabeled data.

The benefits of employing such an approach are: We do not define sequences at the feature level. Rather, by employing a transformer-based [185] approach, similar to BERT [50] we utilize the self-attention layers to attend among patches which does not require big batches of negative samples. Also, the combination of these pretraining tasks result in a significantly faster convergence compared to previous approaches. The resulting representations are evaluated by a scenario that resembles the linear probing evaluation often used in self-supervision [103, 206] and follows the scheme of [2] in the text recognition task. By this assessment, we find that our method outperforms previous self and semi-supervised pipelines. Furthermore, by employing Text-DIAE, we achieve state-of-the-art in handwritten text recognition and document image enhancement, while outperforming scene text recognition under self-supervision settings. The essential findings and novelties of our work are based on the following interesting deductions:

- The impact and combination of pretext tasks depend on the downstream task.
- The closer the association between a pretext task and a downstream task, the better is the model performance.
- By employing Text-DIAE, we achieve faster convergence and use order of magnitude lesser data during pretraining than the contrastive-learning-based approaches.

To add on top of this, this is the first work to our knowledge that investigates different self-supervised pretext tasks for multiple significant downstream tasks in text recognition (HTR-word level, STR) and document image enhancement (document binarization, deblurring) while achieving state-of-the-art performance with 43 and 45 times lesser data for HTR and STR, respectively.

## 9.2 Method

In this section, we present our proposed method for text image recognition and enhancement by describing its building blocks. Our approach uses two steps: a pre-training stage to learn useful representations from unlabeled data and a supervised fine-tuning phase for the desired downstream task.

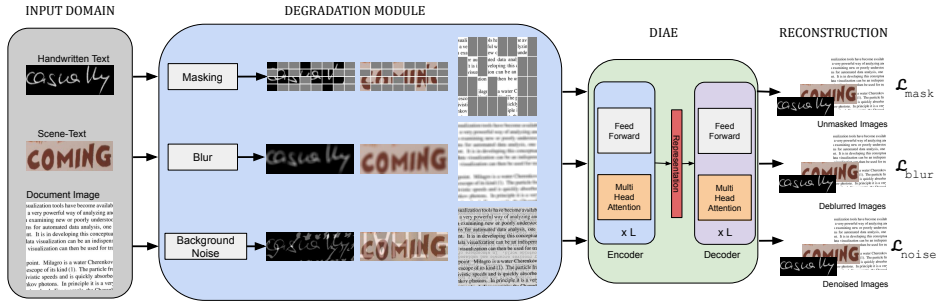


Figure 9.2: **pretraining pipeline.** Text-DIAE aims to learn degradation invariant representations. These are later used to reconstruct the input image with a specific learning objective for each degradation type.

## 9.2.1 pretraining Module

The overall pretraining pipeline of Text-DIAE is shown in Fig. 9.2. For each task, given an unlabeled image  $I$  (eg. a cropped handwritten text, cropped scene text, or a scanned document image) we use a function  $\phi$  to map  $I$  to a degraded form. The function  $\phi$  takes as parameters the original image  $I$  and the degradation type  $\mathcal{T} \in \{mask, blur, noise\}$  where we denote a degraded image by  $I_d = \phi(I, \mathcal{T})$ .

Our model is composed of an encoder  $\mathcal{E}$  and a decoder  $\mathcal{D}$  with learnable parameters  $\theta_{\mathcal{E}}$ ,  $\theta_{\mathcal{D}}$  respectively. The pretraining pipeline trains an encoder function  $\mathcal{E}$  that maps the degraded image  $I_d$  to a latent representation  $z_{\mathcal{T}}$  in a multi-task fashion (unmasking, deblurring, and denoising) and then learning a decoder  $\mathcal{D}$  to reconstruct the original image  $I$  from the representation  $z_{\mathcal{T}}$ :

$$\begin{aligned} z_{\mathcal{T}} &= \mathcal{E}(\phi(I, \mathcal{T}); \theta_E) \\ I_r &= \mathcal{D}(z_{\mathcal{T}}; \theta_D) \end{aligned} \quad (9.1)$$

The learned visual representations from the latent subspace should be invariant to the applied degradation  $\mathcal{T}$ .

**Encoder.** The encoder architecture consists of a vanilla ViT [55] backbone. Given an input image  $I_d$ , it is first split into a set of  $N$  patches,  $I_d^p = \{I_d^{p1}, I_d^{p2}, \dots, I_d^{pN}\}$ . Then, these patches are embedded with a trainable linear projection layer  $E$ . Text-DIAE uses a distinct linear projection layer for every defined pre-text task. These tokens are later concatenated with their 2-D positional information embedded with  $E_{pos}$  and fed to  $L$  transformer blocks to map these tokens to the encoded latent representation  $z_l$ . These blocks are composed of  $L$  layers of Multi-head Self-Attention (MSA) and a feedforward Multi-Layered Perceptron (MLP) as depicted in Figure 9.2. Each of these blocks are

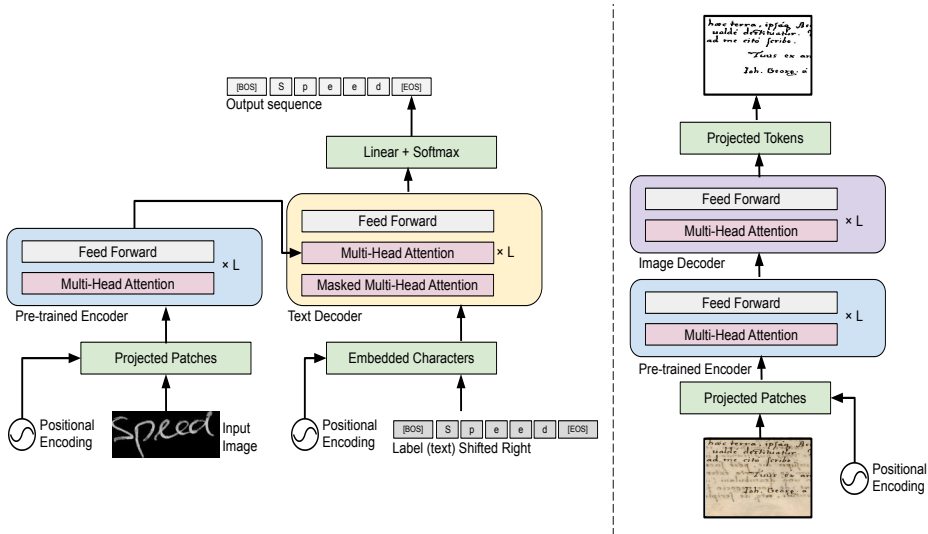


Figure 9.3: **Fine-tuning pipeline.** We start from a pretrained encoder as initial weights to solve a specific downstream task. Explicit decoders are used for text recognition (left) and document image enhancement (right).

preceded by a LayerNorm (LN) [10] and followed by a residual connection:

$$\begin{aligned}
 z_0 &= E(I_d^p) + E_{pos} \\
 z'_l &= \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1, \dots, L \\
 z_l &= \text{MLP}(\text{LN}(z'_l)) + z'_l, \quad l = 1, \dots, L \\
 z_{\mathcal{F}} &= \text{LN}(z_L)
 \end{aligned} \tag{9.2}$$

**Decoder.** The decoder is composed of transformer blocks following the same structure and number of layers as the encoder. The decoder input is the output of encoder  $z_{\mathcal{F}}$ . The output of the decoder is a set of vectors  $I_r = \{I_r^{p1}, I_r^{p2}, \dots, I_r^{pN}\}$  where each of which corresponds to a flattened patch in the predicted (reconstructed) image. Same as before, a distinct linear layer is used for each pre-text task.

$$\begin{aligned}
 z'_l &= \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1, \dots, L \\
 z_l &= \text{MLP}(\text{LN}(z'_l)) + z'_l, \quad l = 1, \dots, L \\
 I_r &= \text{Linear}(z_L)
 \end{aligned} \tag{9.3}$$

## 9.2.2 Fine-Tuning

Our fine-tuning process is illustrated in Fig. 9.3 where we perform two different downstream tasks; text recognition and document image enhancement.



**Text Recognition.** Text recognition aims to transform an image into a machine-encoded form, i.e., a sequence of characters. Let  $I$  be a cropped text image and  $C = \{c_1, c_2, \dots, c_N\}$  its ground truth label which corresponds to a sequence of characters, where  $N$  is the length of the text. The training is done by passing  $I$  to an encoder function  $\mathcal{E}$  to produce a latent representation  $z$ . Then,  $z$  is later fed to a decoder function  $\mathcal{D}'$  to produce a sequence of characters  $C_p = \{c_{p_1}, c_{p_2}, \dots, c_{p_N}\}$  that should match the ground truth label sequence.

We initialize the encoder with the pretrained weights  $\theta_{\mathcal{E}}$  while we employ a sequential transformer decoder [185] as seen in Fig. 9.3-Left. The decoder is initialized randomly and composed of  $L$  transformer blocks of MSA, MLP and Masked-MSA layers preceded by LN layers and followed by a residual connection. The output of the decoder is a sequence of characters where at each time step  $t$ , the predicted character is formed by attending to the representation  $z$  and previous character embeddings until  $t - 1$ .

**Document Image Enhancement.** Document enhancement consists of mapping a degraded document into a clean form. Let  $I_d$  be a degraded image and  $I_c$  its clean version, then the goal is to learn an encoder function  $\mathcal{E}$  that maps  $I_d$  to a representation  $z$  in the same way as in Eqn 9.2.  $\mathcal{E}$  weights are initialized from the pretraining stage. The decoder  $\mathcal{D}''$  generates the clean image  $I_c$  from  $z$  as in Eqn 9.3.

### 9.2.3 Learning Objectives

Our model makes use of different sets of losses for each phase. During pretraining, we use three different losses. Each one is dedicated to a particular pre-text task:  $\mathcal{L}_{mask}$ ,  $\mathcal{L}_{blur}$  and  $\mathcal{L}_{noise}$ . Each of these losses is a mean squared error (MSE) between the reconstructed image  $I_r$  (from the masked, blurred, or noisy image) and its ground-truth version  $I_{gt}$ . Thus, the overall loss for our pretraining stage is:

$$\mathcal{L}_{pt} = \lambda_1 \mathcal{L}_m(I_r, I_{gt}) + \lambda_2 \mathcal{L}_b(I_r, I_{gt}) + \lambda_3 \mathcal{L}_n(I_r, I_{gt}) \quad (9.4)$$

Where during our experimentation, the best results were obtained with setting  $\lambda_1 = \lambda_2 = \lambda_3 = 1$ . Also, while fine-tuning on text recognition, we use a cross-entropy loss between the predicted sequence of characters  $C_p$  and  $C$ . And document image enhancement fine-tuning, we used an MSE loss between the cleaned image  $I_c$  and  $I$ .

## 9.3 Experiments

In this section we describe the studied scenarios and experiments performed for text recognition and document enhancement respectively. We ask the reader to refer to the supplementary material for specific implementation details.

Table 9.1: **Representation quality.** We evaluate the encoder capability of learning visual representations. This scenario is analogous as the linear probing in self-supervised models. We train a decoder with labelled data on top of a frozen encoder pre-trained on the proposed degradation. The column *Seen* refers to the number of samples in millions seen during pre-training. Word prediction in terms of Accuracy (Acc) and single edit distance (ED1) in handwritten and text recognition.

Method	Encoder	Decoder	Handwritten Text						Scene-Text					
			IAM			CVL			IIIT5K			IC13		
			Acc	ED1	Seen	Acc	ED1	Seen	Acc	ED1	Seen	Acc	ED1	Seen
simCLR [34]			4.0	16.0	205.8	1.8	11.1	205.8	0.3	3.1	409.6	0.3	5.0	409.6
seqCLR [2]	CNN	CTC	39.7	63.3	205.8	66.7	77.0	205.8	35.7	62.0	409.6	43.5	67.9	409.6
PerSec [120]			-	-	-	-	-	-	37.9	-	-	46.4	-	-
PerSec [120]	ViT		-	-	-	-	-	-	38.4	-	-	46.7	-	-
simCLR [34]			16.0	21.2	205.8	26.7	30.6	205.8	2.4	3.6	409.6	3.1	4.9	409.6
seqCLR [2]	CNN	Attn.	51.9	65.0	205.8	74.5	77.1	205.8	49.2	68.6	409.6	59.3	77.1	409.6
PerSec [120]			-	-	-	-	-	-	50.7	-	-	61.1	-	-
PerSec [120]	ViT		-	-	-	-	-	-	52.3	-	-	62.3	-	-
<b>Ours</b>	ViT	Transf.	<b>71.0</b>	<b>82.1</b>	<b>4.7</b>	<b>78.1</b>	<b>81.5</b>	<b>1.2</b>	<b>77.1</b>	<b>87.8</b>	<b>9.1</b>	<b>92.6</b>	<b>95.6</b>	<b>18.2</b>

### 9.3.1 Text Recognition

**Evaluating Representations.** In order to evaluate the quality of the learned representations, and extend commonly used linear-probing settings [206], we employ a similar approach as introduced by [2]. As a first step, the encoder is pretrained with unlabeled data as described in Section 9.2.1. After that, the encoder’s weights are frozen and a new decoder is trained on top of it with all the labeled data. The decoder, as we detailed above, generates the predicted characters in a time-step manner. Since the encoder remains frozen, this scenario is a good proxy that represents the expressivity of the learned visual representations. To this end, Table 9.1 shows the results of our proposed approach. We compare self-supervised methods specifically designed for the text recognition task.

**Better performance.** As it can be seen from Table 9.1, the seqCLR method presented by [2] improves significantly a self-supervised baseline inspired by SimCLR [34]. In the recently released approach PerSec by [120], they slightly improve over the seqCLR. It is evident that our Text-DIAE model *greatly* outperforms all the aforementioned state-of-the-art approaches regarding the representation quality obtained, both in handwritten and scene-text. The improvements in terms of the accuracy in a handwritten text dataset, IAM, is close to **+20 points**. Moreover, a bigger improvement gap is obtained when evaluating scene text. An average gain of **+30 points** is accomplished in IIIT5K and ICDAR13, proving the generalization of our method to different domains. In our model, the great expressivity of features achieved by the encoder is mainly due to two factors. Firstly, by masking image patches, the encoder learns a strong unigram character distribution (refer to Figure 9.4), which is not leveraged in previous methods. Secondly, by distorting and recovering the image, we make the model learn richer rep-

Table 9.2: **Semi-supervised results.** Accuracy obtained by fine-tuning a pre-trained model with varying percentages of the labeled dataset. Under this setting, we back-propagate the gradients through the specific decoder and the pre-trained encoder.

Method	Encoder	Decoder	Handwritten Text						Scene-Text	
			IAM			CVL			IIIT5K	IC13
			5%	10%	100%	5%	10%	100%	100%	100%
Supervised [2]	CNN	CTC	21.4	33.6	75.2	48.7	63.6	75.6	76.1	84.3
simCLR [34]			15.4	21.8	65.0	52.1	62.0	74.1	69.1	79.4
seqCLR [2]			31.2	44.9	76.7	66.0	71.0	77.0	80.9	86.3
PerSec [120]			-	-	77.9	-	-	78.1	82.2	87.9
PerSec [120]			ViT	-	-	78.0	-	-	78.8	83.7
Supervised [2]	CNN	Attn.	25.7	42.5	77.8	64.0	72.1	77.2	83.8	88.1
simCLR [34]			22.7	32.2	70.7	59.0	65.6	75.7	77.8	84.9
seqCLR [2]			40.3	52.3	79.9	<b>73.1</b>	<b>74.8</b>	77.8	82.9	87.9
PerSec [120]			-	-	80.8	-	-	80.2	84.2	88.9
PerSec [120]			ViT	-	-	<b>81.8</b>	-	-	80.8	85.2
Supervised (Ours)	ViT	Transf.	22.8	25.3	71.7	17.9	19.8	71.9	75.7	91.9
<b>Ours</b>			<b>49.6</b>	<b>58.7</b>	80.0	47.9	68.5	<b>87.3</b>	<b>86.1</b>	<b>92.0</b>

representations to detect and recover the text into a clean and readable state. Thus, the model is learning the most valuable features that lead to the best recognition performance.

**Faster convergence.** One of the most important outcomes by employing our method, is that a **paramount** improvement in convergence is achieved during pretraining. Table 9.1 shows this effect under the column labelled as “Seen”. It depicts the total number of seen samples that each model requires during the pretraining stage. It is worth highlighting that during pretraining the encoder of our model requires **43** and **166** times lesser data in IAM and CVL respectively when compared to the seqCLR and simCLR. In scene-text, our model employs only 18.2M samples to yield powerful representations compared to the 409M samples required by previous self-supervised approaches.

**Fine-Tuning.** In this stage, we evaluate our model considering a semi-supervised setting where the obtained results are depicted in Table 9.2. Here we use the self-supervised pretrained encoder as a backbone and train a transformer-based decoder from scratch that predicts the characters in a sequential manner, as illustrated in Fig. 9.3-Left. In this scenario, the gradients are back-propagated not only to the decoder but also to the encoder. Following the previous work [2], we use 5% and 10% of the labeled dataset by randomly selecting the training samples. As suggested in [34] we perform fine-tuning on all the labeled dataset. In order to compare with [2] and since the scene-text dataset is synthetic, we evaluate with the complete labeled dataset.

**Higher performance in fine-tuning settings.** Our model exploits data in a more efficient manner than previous self-supervised methods in the fine-tuning setting. We

Table 9.3: **Ablations of the pre-training objectives.** Results in handwritten and scene-text recognition obtained by each pretext task. The performance is measured in terms of Word and Character error rates (WER and CER).

$\mathcal{L}_{mask}$	$\mathcal{L}_{blur}$	$\mathcal{L}_{noise}$	IAM			IC13		
			CER↓	WER↓	Avg.	CER↓	WER↓	Avg.
✓	✗	✗	<b>9.3</b>	<b>20.0</b>	<b>14.65</b>	4.5	8.0	6.25
✓	✓	✗	12.3	24.8	18.5	<b>4.2</b>	<b>8.0</b>	<b>6.10</b>
✓	✗	✓	11.1	23.3	17.2	4.8	8.6	6.70
✓	✓	✓	11.4	23.8	17.6	5.1	9.3	7.20

infer that the set of degradations proposed yields rich signals, helping the encoder to adapt to the downstream task more efficiently. Our model achieves state-of-the-art in all scenarios when all the labeled datasets are used except in IAM where the PerSec is slightly better. Under semi-supervised settings, our model performs better at the IAM dataset when employing 5% and 10% of the labels than simCLR and seqCLR. Since CVL contains substantially fewer data samples than IAM, SeqCLR still outperforms our approach in the CVL dataset. However, while employing the full labels of CVL, Text-DIAE outperforms all the methods by a large margin.

**More efficient than a supervised baseline.** From table 9.2, we can also notice the superiority of pretraining our architecture compared to a fully supervised model starting from scratch. This suggests that the self-supervised pretraining of such transformer-based architectures is essential to obtain better results, and beneficial especially in small labeled datasets scenarios, since the unlabeled data is generally easier to obtain for a self-supervised pretraining.

**The effect of fine-tuning after pretraining.** By proposing the degradation invariant optimization at pretraining, our model achieves a significant gain in recognition on handwritten text datasets. An average of 10 points of accuracy are gained after fine-tuning (refer to Table 9.1 and 9.2). Finally, it is important to note that our model reaches state-of-the-art in the handwritten text recognition task, even compared to specifically designed supervised approaches. The results on the IAM dataset are shown in Table 9.4, which measures the performance of a model in terms of word and character error rate, WER and CER respectively.

**Ablation Studies.** The results of experimentation regarding the effect of each degradation as a pretext task in pretraining are given in Table 9.3. Firstly, among the three proposed degradations, masking is the most crucial to be applied in both tasks, handwritten and scene text recognition. When an input word is masked, and in order to properly reconstruct it, the model has to learn a character level distribution. This by itself provides a strong prior compared to denoising or deblurring an image. Additionally, adding blur in scene-text imagery improves the representations learned by the model shown by the results. Lastly, adding noise does not result in an improvement in text recognition tasks. However, as it is shown in the next section, the combination of



Figure 9.4: **Qualitative results of pretraining samples.** The left refers to handwritten text, while scene-text is depicted on the right. On each scenario, from left to right, the original, masked and reconstructed images are depicted.

Table 9.4: **SOTA results.** Quantitative evaluation with state-of-the-art methods on the IAM word level dataset.

Method	CER↓	WER↓	Avg.
Bluche et al. [23]	7.3	24.7	16.00
Bluche et al. [24]	7.9	24.6	16.25
Sueiras et al. [178]	8.8	23.8	16.30
ScrabbleGAN [59]	-	23.6	-
SSDAN [210]	8.5	22.2	15.35
SeqCLR [2]	9.5	20.1	14.80
PerSec [120]	-	18.2	-
<b>Ours</b>	<b>9.3</b>	<b>20.0</b>	<b>14.65</b>

the 3 degradation produces a richer encoder in document enhancement. Therefore, we can safely assume that each degradation has a task-dependent impact on the representations learned depending on their similarity of them when compared to the final downstream task and input data distribution.

**Qualitative Results.** In Figure 9.4 we show the reconstructed images at pretraining stage for handwritten and scene-text samples. It is important to note the complexity of the reconstruction task even for humans. Even though high masking percentages are employed (75%), our model learns to properly adapt to handwritten styles and fonts found in scene text. As can be appreciated, although sometimes our model’s reconstruction does not match with the ground truth images, it can still reconstruct the most probable and plausible English words (e.g. see “school” vs “sand” in 4th row in handwritten examples). Another interesting outcome is also noticed for the scene-text example where “xperia” is reconstructed correctly while the last character “a” is selected from another font, demonstrating the model’s capability. Minor reconstruction errors are found such as that the model eventually learns to overcome at the fine-tuning stage.

### 9.3.2 Document Image Enhancement

**Performance Analysis on Binarization.** As shown in Table 9.5, the Text-DIAE outperforms the previous state-of-the-art approaches on the majority of the standard metrics for document binarization tasks. Specifically, the quantitative comparison of results demonstrates that Text-DIAE achieves an optimal gain in PSNR, FM,  $F_{ps}$  and DRD performance surpassing all previous arts. The largest performance improvement is obtained over the H-DIBCO 2012 while the least performance gain is obtained in the H-DIBCO 2018. One of the major concerns that degraded historical documents face is the show-through effect, which appears when ink impressions from one side of the document start appearing on the other side, making it almost impossible to read. The enhanced Text-DIAE output illustrates that it not only resolves the show-through but also sharpens and smoothens the edges of the foreground text approximately to the ground-truth level.

Table 9.5: **SOTA results.** Comparison of the proposed Text-DIAE compared to previous state-of-the-art approaches on the different DIBCO and H-DIBCO Benchmarks

Method	DIBCO Benchmarks															
	2011				2012				2017				2018			
	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}\uparrow$	DRD $\downarrow$	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}\uparrow$	DRD $\downarrow$	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}\uparrow$	DRD $\downarrow$	PSNR $\uparrow$	FM $\uparrow$	$F_{ps}\uparrow$	DRD $\downarrow$
[164]	15.60	82.10	-	8.50	16.71	82.89	87.95	6.59	14.25	77.11	84.1	8.85	13.78	67.81	74.08	17.69
[96]	19.90	<b>95.50</b>	-	1.80	21.37	95.16	96.44	1.13	15.85	91.57	93.55	2.92	19.39	89.71	91.62	<b>2.51</b>
[211]	20.30	93.80	-	1.80	21.91	94.96	96.15	1.55	17.83	90.73	92.58	3.58	18.37	87.73	90.60	4.58
[170]	20.81	94.37	96.15	1.63	22.29	95.31	96.29	1.60	19.11	92.53	95.15	2.37	19.46	90.59	93.97	3.35
<b>Ours</b>	<b>21.29</b>	95.01	<b>96.86</b>	<b>1.48</b>	<b>23.66</b>	<b>96.52</b>	<b>97.04</b>	<b>1.10</b>	<b>19.64</b>	<b>93.84</b>	<b>95.71</b>	<b>1.93</b>	<b>19.95</b>	<b>91.32</b>	<b>94.44</b>	3.21

**Performance Analysis on Deblurring.** In Table 9.6 we show a quantitative comparison and superiority of Text-DIAE over supervised techniques [80, 191, 174, 170] on the document deblurring benchmark. A substantial gain in PSNR by **+2 points** on a **logarithmic** scale is obtained over DocEnTr [170], which signifies the greater quality of deblurred images generated by Text-DIAE. There are two different kinds of blurring which appear in documents: motion blur owing to the sudden rapid camera movement and out-of-focus blur which emerges when the light fails to converge in the image. In Fig. 9.5, we show an interesting qualitative case study of a motion-blurred document image. We assess the performance of deblurring by running the Tesseract-OCR engine [168] over the blurred, ground-truth, DocEnTr prediction and the Text-DIAE output. Qualitative results show that Text-DIAE significantly decreases the CER, showing vast improvement in OCR performance as depicted in green font.

**Ablation Studies.** We also showcase an interesting ablation on the task of document image binarization for the challenging DIBCO 2018 benchmark. From Table 9.7, we infer that any pretraining task is beneficial while the denoising task is the most crucial to be applied when each pre-text task is applied separately. The aforementioned result can be attributed to the fact that denoising is much closer to the downstream binarization task. Also, it demonstrates that Text-DIAE performs the best for document enhancement tasks when the model learns all the possible degradation (masking, blurring, and adding noise) together.

Table 9.6: **SOTA results:** Quantitative evaluation with state-of-the-art methods on the deblurring dataset.

Method	PSNR
CNN-Baseline [80]	19.36
Pix2Pix-HD [191]	19.89
DE-GAN [174]	20.37
DocEnTr [170]	21.28
<b>Ours</b>	<b>23.58</b>

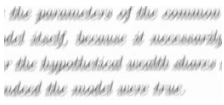
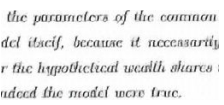
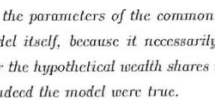
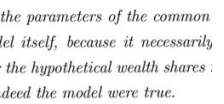
Original Input	DocEnTr [170]	Ours	Ground Truth
			
OCR output: "Mae yw spaniedod"» if MA AAPAIMAPE dosh anelf, Awanaisnn A dnmoupil l Mie Myrtle dial cell sagos Alo Wie sasclde saesyie dias,"	OCR output: "the parameters of Ure commnon del iticif, because it mecensartly r the hypotictical wealth shares ndeed the model worn truc."	OCR output: "the parameters of the common del ituelif, because it necessarit r the hypothetical wealth shares rdeed the model were true."	OCR output: "the parameters of the common del itself, because it necessarily r the hypothetical wealth shares ndeed the model were true."
CER: 78.86	CER: 18.51	CER: <b>8.94</b>	CER: 4.88

Figure 9.5: **Qualitative results of deblurred samples.** The document image on the left refers to the originally captured blurred image, followed by the ground-truth, and the deblurred results from the DocEnTr and our Text-DIAE model towards the right. The correctly predicted OCR output is shown in "Green" font while the inaccurate ones are depicted in "Red" and recognition performance in terms of CER.Table 9.7: **Ablations of the degradations as pre-training objectives.** Results in document image binarization on DIBCO 2018 obtained by each pretext task in terms of PSNR.

$\mathcal{L}_{mask}$	$\mathcal{L}_{blur}$	$\mathcal{L}_{noise}$	PSNR
✗	✗	✗	18.75
✓	✗	✗	19.65
✗	✓	✗	18.98
✗	✗	✓	19.82
✗	✓	✓	19.34
✓	✗	✓	19.45
✓	✓	✓	<b>19.95</b>

## 9.4 Conclusion

This work demonstrates the capability of learning richer representations through pretext degradation tasks. Self-supervised learning can immensely boost the performance of text recognition and document image enhancement without any requirement of labeled data. Notably, we show that Text-DIAE does not share the limitations of contrastive or sequential approaches and is more effective at learning rich representations while seeing *significantly* fewer data points. Extensive experimentation during fine-tuning demonstrates that Text-DIAE surpasses previous supervised and self-supervised state-of-the-art in handwritten text recognition and document image enhancement while outperforming previous self-supervised approaches in scene-text recognition. We hypothesize that Text-DIAE performs complex variable reconstructions during pre-training, which helps to learn meaningful visual concepts from the latent representation space. We also provide the community with the following insights to work on: 1) Designing new pretext tasks that are similar to downstream tasks. 2) The effect/trade-off of a combination of various pretext tasks on the downstream tasks. 3) A need for a holistic approach to combine all the tasks into a single model.





## **Part III**

# **User Evaluation of HTR Systems in Low Resource Data**



# Chapter 10

## Evaluation of HTR systems for the Automatic Transcription of Rare Manuscripts from a User perspective: Application to *Codex Runicus*

---

*Recent breakthroughs in Artificial Intelligence, Deep Learning, and Document Image Analysis and Recognition have significantly eased the creation of digital libraries and the transcription of historical documents. However, for documents in rare scripts with few labeled training data available, current Handwritten Text Recognition (HTR) systems are too constrained. Moreover, research on HTR often focuses on technical aspects only, and rarely puts emphasis on implementing software tools for scholars in Humanities. In this Chapter, we describe, compare and analyze different transcription methods for rare scripts. We evaluate their performance in a real use case of a medieval manuscript written in the runic script (Codex Runicus) and discuss the advantages and disadvantages of each method from the user perspective. From this exhaustive analysis and comparison with a fully manual transcription, we raise conclusions and provide recommendations to scholars interested in using automatic transcription tools.*

---

## 10.1 Introduction

The protection and preservation of cultural heritage has been considered of paramount importance for decades. This awareness was especially evident after the two World Wars and increased with the UNESCO World Heritage Convention in 1972. Among the cultural assets and objects (e.g. archaeological sites, buildings, paintings, coins, etc.), historical documents form one important part containing valuable information related to the culture and memory of our past societies. There have been many digitization campaigns in archives and libraries around the world. However, this is just the first step towards the spread of our cultural heritage. Indeed, access to the information contained in document collections remains limited until the documents are properly transcribed, indexed, or even, linked.

Manual transcription of a vast amount of historical documents is extremely time-consuming and their evaluation requires the interdisciplinary expertise of scholars in paleography, history, etc. Recent breakthroughs in Artificial Intelligence, Deep Learning, Computer Vision, and Document Image Analysis and Recognition, in particular, have eased the processing of documents for creating digital libraries. In fact, current deep-learning-based Handwritten Text Recognition (HTR) methods obtain satisfactory performance, so that, in theory, scholars can significantly speed up the transcription process.

However, there are some practical limitations. First, research on HTR technology has often focused on algorithms and methods from a mere technical perspective, frequently disregarding the development of software tools for end users, such as scholars in Humanities. Consequently, scholars require a substantial technical background for applying such techniques to their particular manuscripts. Second, deep-learning algorithms need a significant amount of labeled data to train. This constraint poses a problem when transcribing uncommon scripts or alphabets (e.g. cuneiform, runes, ciphered documents, Egyptian hieroglyphs, etc.) [172] because, contrary to common scripts (e.g. Latin, Chinese, Arabic), such labeled data is barely available. Indeed, existing methods are often too constrained since they need labeled data for training and fine-tuning. As a result, in practice, most of the existing HTR methods have little applicability for transcribing uncommon scripts.

Therefore, our goal is to research and develop useful computational methods and software tools to facilitate transcribing rare scripts for which no or very limited amount of training data can be provided. In this work, we propose a first step towards the creation of such generic tools by developing and evaluating various transcription methods for rare scripts. As a real use case, we apply the transcription methods on the *Codex Runicus*, an Old Danish manuscript text with legal works from the end of the 13th century written in runic script.<sup>1</sup>

---

<sup>1</sup>The runes had been employed by Germanic people since the 2nd century AD and survived in Scandinavia also after the introduction of the Latin alphabet around the year 1000. Runes were mainly used epigraphically, i.e. for inscriptions on solid materials, such as stone or wood. The *Codex Runicus* is one of only two longer manuscript texts that were written in runes.

The contribution of this work is two-fold. Firstly, we have developed four different transcription methods, each one with different characteristics (e.g. required amount of labeled data, segmentation level, speed, etc.). Concretely, one learning-based method is based on recurrent neural networks, one learning-free method based on clustering, and two few-shot learning methods (one for classification and another for detection). Secondly, we have exhaustively evaluated these methods both quantitatively and qualitatively with regard to their performance. We have discussed the advantages and disadvantages of each method as well as their applicability from the user perspective considering time consumption in comparison with a fully manual transcription and opportunities for user validation. On the basis of this evaluation, we give recommendations for further development and usage of transcription tools for this kind of manuscript.

The rest of the Chapter is organized as follows. Section 10.2 reviews the existing HTR methods and software tools. Section 10.3 describes the different methods that have been developed. Section 10.4 analyzes the experiments and Section 10.5 raises recommendations for scholars. Finally, Section 10.6 concludes the Chapter.

## 10.2 Related work

In this section, we review the existing HTR methodologies and software tools.

### 10.2.1 Transcription methods for historical manuscripts

Handwritten text recognition (HTR) is an active research field in computer vision. Nowadays, one may find many different approaches for transcribing popular and widely used scripts, such as Latin [93, 42], Chinese [212], Arabic [89] alphabets. Thanks to the available resources in those alphabets (especially in terms of annotated data), most existing methods opt for data-driven deep learning architectures, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Using those deep learning-based methodologies, a mapping function from the handwritten image to the text characters is learned in a supervised way, leading to very satisfactory performance. Moreover, recognizing text by Keyword Spotting Systems (KWS) was applied for historical text in [161, 162]. The time of using KWS and validating the output by a user was shown to be a better option compared to the fully manual transcription.

However, in the case of uncommon or rare alphabets (e.g. Egyptian hieroglyphs, cuneiform, runes), the literature is not so prolific. Not surprisingly, the few existing approaches are usually learning-free methods, because the performance of deep learning models dramatically decreases when there is few annotated data to train. For this reason, some researchers propose learning-free spotting for cuneiform [158, 25], whereas others propose unsupervised transcription for ciphered manuscripts [14, 202]. Although learning-free methods are flexible, their performance is moderate compared to

learning-based approaches, especially when alphabets contain very similar symbols or when characters are difficult to segment (e.g. touching characters in cursive writing). Consequently, considerable manual intervention is needed to validate the HTR output for an acceptable result.

Lately, and as an alternative to standard deep learning, Few-shot learning has been proposed for computer vision tasks, including image classification [116, 111, 112] and object detection [142]. These approaches require very few annotated data while reaching a performance close to the standard deep learning-based ones. Founded on the way humans learn novel concepts, few-shot has the ability to easily adapt to unseen data. Thus, the data used for training is very few compared to standard deep learning methods [169].

## 10.2.2 Transcription tools and user platforms

While research on HTR methods for historical manuscripts has significantly progressed in recent years, Virtual Research Environments that make the technology accessible and available for the Humanities scholars, archives, or the interested public are still rare. Many transcription tools such as FromThePage<sup>2</sup>, Scripto<sup>3</sup> or eLaborate<sup>4</sup> are designed to facilitate the (collaborative) manual transcription and provide tools to produce digital editions, but do not include HTR technology.

The most popular platform that offers an infrastructure for transcription and text search of historical manuscripts is *Transkribus*. It has a workflow that includes the upload of digitized images, layout analysis tools for segmenting the digitized page images into text lines, and transcription tools for transcribing these lines. The GUI allows both for automatic and manual segmentation to mark lines. Users have the option to transcribe either in a downloadable application or in an online web interface.<sup>5</sup> It is possible to use pre-trained HTR models or train one's own HTR model from the ground truth. The tool outputs a "confidence matrix" where the likelihood of a transcription character is shown in the image, which allows the user to decode the confidence into the final transcribed text. The tool can obtain very good results (a CER lower than 5%) in certain cases (cf. [132]), and includes pre-trained HTR models for Latin, Devanagari, and Cyrillic alphabets. However, a crucial limitation of *Transkribus* is the need for a very large ground truth data set for training (15,000 transcribed words) [132, pp. 959, 967]. This precondition makes the technology unavailable for manuscripts using rare scripts, because such a large data set simply does not exist. An additional challenge for transcribers of rare scripts is that these writing systems may include written characters without a current Unicode coding (e.g. some runes in the here analysed *Codex Runicus* [141]).

Contrary to *Transkribus*, *eScriptorium* [98] is an open-source transcription plat-

<sup>2</sup><https://fromthepage.com> (accessed 20 July 2021)

<sup>3</sup><https://scripto.org/> (accessed 20 July 2021)

<sup>4</sup><https://elaborate.huygens.knaw.nl> (accessed 20 July 2021)

<sup>5</sup><https://transkribus.eu/lite/#features> (accessed 20 July 2021)

form<sup>6</sup>, based on Kraken, an HTR system based on neural networks. The platform also includes tools for layout analysis, and it is specifically designed for transcribing historical manuscripts of any kind or alphabet. However, it has the same limitation as *Transkribus*: the methodology is based on deep learning, so it requires a significant amount of labeled data (aprox. 100 pages) to retrain the models for any new scripts.

The newly released platform *Fabricus*<sup>7</sup>, a tool included as an experiment in Google Arts & Culture, also uses deep learning to translate ancient Egyptian hieroglyphs. It features a workbench for researchers that includes the possibility to upload a digitized image, clean the image, separate the different signs, and automatic recognition, classification and translation of the hieroglyphs. The classification is supported by a trained neural network, whereas for the translation, the hieroglyphs are matched to dictionaries and published translations.<sup>8</sup> Obviously, this tool is restricted to documents in hieroglyphs and cannot be used for other kinds of scripts and writing systems.

In summary, generic and flexible transcription tools that do not imply a significant human effort in providing manually annotated data are still lacking for the end-user that wishes to transcribe manuscripts with rare scripts.

### 10.3 Transcription methods

In this section, we describe the methods that have been developed for transcribing rare scripts. We have selected one representative method of each methodology so that we can analyze the advantages and disadvantages of each methodology when applied to a real use case. First, we have developed a deep learning method based on Long Short-term Recurrent Neural Networks (LSTM-RNNs), which is very similar to the HTR methods used in *Transkribus* and *eScriptorium*. Second, we have developed an unsupervised clustering method (Cluster(sup)), and included a semi-supervised variant that benefits from the user feedback (Cluster(semi-sup)). Third, we have adapted a few-shot learning method proposed for image classification (FS-classification) to our transcription task. Finally, we have developed a few-shot learning method originally proposed for object detection (FS-detection), and applied it to transcription.

Table 10.1 describes the characteristics of each method. The user effort refers to the amount of labeled data (ground-truth) that the user must provide to each method. The segmentation level indicates whether the method requires, as input, segmented lines or symbols. The performance indicates the typical accuracy, in general (as expected, learning-based methods are usually the best). Scalability refers to the generalization ability of the method with respect to the different kinds of manuscripts. Finally, the hardware needs refer to the memory and processing power needed to train the differ-

<sup>6</sup><https://escripta.hypotheses.org/> (accessed 20 July 2021)

<sup>7</sup><https://artsexperiments.withgoogle.com/fabricus/en> (accessed 20 July 2021)

<sup>8</sup>The platform additionally engages the wider public with two features: a "Learn" tool with tutorials and training sequences and a "Play" feature where one can create their own messages in hieroglyphs and share them.



ent methods. As it can be observed, there is no method that excels in all aspects.

Method	User effort (labelled data)	Segmentation level	Performance	Scalability	Hardware needs
LSTM-RNNs	High	Line	High	High	Medium
Cluster(sup)	None	Symbol	Moderate	Low	Low
Cluster(semi-sup)	Very Low	Symbol	Moderate	Medium	Low
FS-classification	Low	Symbol	Good	Medium	Medium
FS-detection	Low	Line	Good	High	Medium

Table 10.1: Overview of the characteristics of the different methodologies.

Next, we describe the preprocessing steps commonly applied before the transcription step. Then, we will describe the four transcription methods.

### 10.3.1 Preprocessing: Binarization and Segmentation

First of all, the manuscript is scanned at a minimum of 200 dpi, and the image is binarized using Sauvola’s adaptive binarization method [164]. The aim of the binarization is to remove the background noise as much as possible, highlighting the text as foreground pixels. Afterward, vertical and horizontal projections are used to segment the text in the page and disregard the page margins. Then, text lines are segmented using horizontal projections.

The above-described preprocessing steps are common to all the methods. However, the clustering and few-shot classification methods need to segment the text lines into symbols. So, to fulfill the segmentation requirements of these latter methods, we segment the symbols as follows. First, we compute the connected components in the image to isolate the symbols. Then, and since some symbols are formed by several connected components (e.g. symbols with dots and accents), we group the connected components according to their distance and center of mass to obtain the final character segmentation.

Since our objective is to analyze the different transcription algorithms, we have manually checked and corrected any inaccuracy in the segmented lines and symbols. Thus, we make sure that the segmentation does not affect the transcription experiments.

### 10.3.2 Learning-based method: LSTM-RNNs

The first implemented transcription method is a deep learning-based approach based on Long Short-term Memory blocks Recurrent Neural Networks (LSTM-RNNs), which is one of the most well-known and widely used approaches for handwritten text recognition (HTR). In the case of handwriting recognition, the input is an image of a textline,

whereas the output is the sequence of transcribed characters, which is typically obtained using Bi-directional LSTM-RNNs [62]. It consists of computing a (previously defined) vector of features for each column in the image, and processing the textline in both directions (left-to-right and right-to-left) with the recurrent networks. In this architecture, the Connectionist Temporal Classification (CTC) loss is used for training.

Lately, several architecture improvements have been proposed. Concretely, we have opted for Multi-dimensional LSTMs (MDLSTM) [153, 60] because they obtain good results while avoiding computing a manually pre-defined feature vector over the image. In this way, the network can automatically learn which is the most suitable feature representation for each particular alphabet or manuscript. However, it is a deep learning-based method, so it needs a considerably amount of labelled data to properly learn.

### 10.3.3 Learning-free method: Unsupervised Clustering

In this Section, we describe the unsupervised and semi-supervised clustering methods. Clustering is typically used for classification tasks, and has been adapted for transcribing ciphered texts [14]. In this case, the input of the method is the set of segmented symbols in the manuscript.

#### Method description.

The method is composed of two steps: hierarchical clustering and label propagation. First, we compute the SIFT [166] descriptor for each segmented symbol image, and apply a hierarchical k-means [7] to obtain clusters. At the beginning, the algorithm assumes that all symbols belong to the same cluster. Then, each cluster is recursively subdivided into smaller clusters until the clusters are no more divisible or because we have reached the minimum amount of symbols per cluster. Then, the most populated clusters are automatically selected to be used as initial seeds for the label propagation algorithm. The objective is to progressively propagate the labels (e.g. the cluster id) of these populated clusters through the rest of the (not yet labeled) symbols. The algorithm repeatedly assigns a label to each unlabelled symbol according to the labels of its neighbors, until convergence (i.e. no more label changes).

Except for the initial seeds, the assignation of labels is a soft assignment, which means that each symbol has a vector of probabilities between 0 and 1 for each possible label (e.g. an element  $x$  has a probability of 65% of belonging to cluster 4, and a 35% of belonging to cluster 7). To determine the final label for each symbol, we set a confidence threshold of 60% as it was shown to be the optimal choice in [14]. For each symbol, if the most probable label has a higher probability than a given threshold, we will assign that label to the symbol. Otherwise, that symbol will remain unlabelled because there is no consensus (of course, it is better to leave a symbol as unlabelled rather than assigning a wrong label). We will use the label \*/? for those unlabelled/unknown

symbols.

### Modalities.

In an ideal situation, each initial seed for the label propagation should have instances of the same symbol class. However, it is common to find symbols in the alphabet that are visually similar. Consequently, a cluster may contain samples from different symbol classes as can be seen in Fig. 10.1, so we may propagate wrong label information. For this reason, we have designed the following modalities:

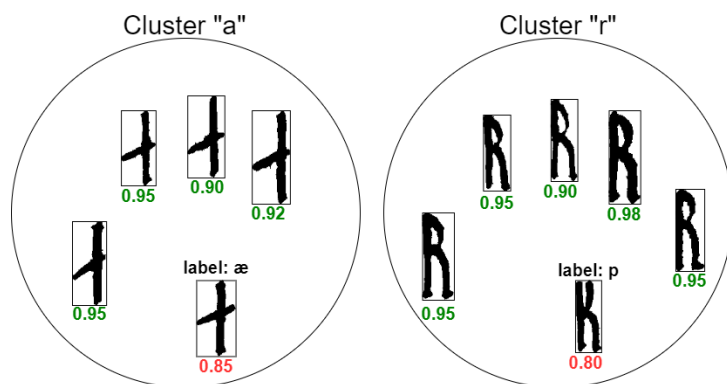


Figure 10.1: Examples of elements from two different clusters. The scores shown in green mean that the element is correctly clustered, while the ones in red are wrongly clustered because of the high visual similarity.

**Without user intervention** In this case, there is no user intervention. The labels in each cluster are propagated. So the user just validates and corrects the transcription errors at the end of the process.

**With user intervention** In this scenario, the user can correct the clusters before the label propagation starts, by doing two interventions:

- Cleaning the initial seeds. The user removes those symbols that do not belong to that cluster (see figure 10.2a)). Since we ensure that the initial labels to propagate are correct, the final transcription error is minimized.
- Select a cluster for each different symbol. The user can assign a cluster for each symbol in the alphabet (see figure 10.2b)). In this way, we can ensure that the labels to be propagated cover all the alphabet of symbols.

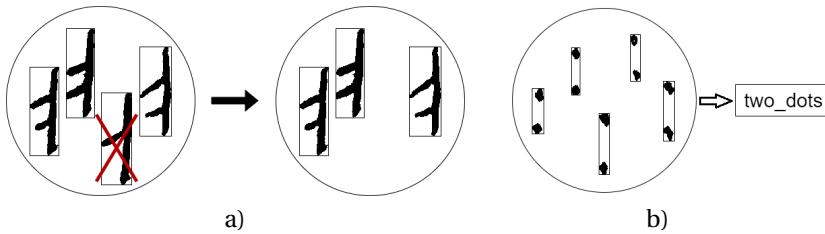


Figure 10.2: Unsupervised clustering with user intervention. Left: The user removes the wrongly clustered symbol in each cluster. Right: The user assigns the corresponding label/transcription to each cluster.

### 10.3.4 Few-Shot Classification method

In this section, we describe the first few-shot learning method that we have developed for transcription, which is inspired by the few-shot learning architecture presented in [163] and the extensions proposed in [16]. Graphs have always been a powerful tool to represent a set and the relationships between its elements, hence it was evident to make use of this in Graph Convolutional Networks (GCNs). The main idea is to represent each symbol as a node in the graph, and learn the similarity degree between each pair of symbols, stored in the edges of the graph. Contrary to the previous deep learning method, few-shot learning is able to learn in a particular set of classes and test in a completely new set of classes. This means that the method can be trained on common scripts (with training data available), and applied to unseen scripts.

#### Method description.

The few-shot architecture for symbol classification assumes that the input is segmented symbols. Then, Convolutional Neural Networks are used to learn the embedding for each node in the graph. At the beginning, it is the one-hot encoding of the label if the label is known, and a uniform probability distribution otherwise. Ideally, and once trained, the model is able to distinguish among symbol classes never seen before. We have improved the original architecture with some improvements proposed in [16]. The first one consisted of exploring the information available in the edges, which resulted in the omission of the model's last layer and a decrease in the number of trainable parameters. This model is illustrated in Figure 10.3.

#### Adaptation for transcription.

We concatenate the label for each segmented symbol to obtain the final transcription. If we have  $k$  examples for each one of the  $N$  different symbols in the alphabet, the problem is considered an  $N$ -way,  $k$ -shot classification task. In the experiments, we

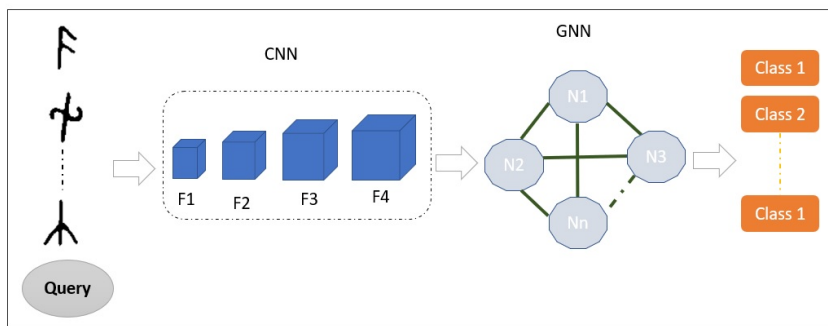


Figure 10.3: Few-Shot classification model.

have used the same number of examples per symbol class, this means that, during the transcription, the number of ways and shots does not change.

In the original few-shot classification, the experiments are performed over disjoint sets, which means that the training, validation, and test sets contain a distinct set of classes. However, for transcription, we require to classify all the symbol classes in the alphabet. For this reason, our training, validation, and test sets contain instances of each one of the symbols in the alphabet.

### 10.3.5 Few-shot Detection method

The previous few-shot method requires segmenting the text line into isolated characters. Since the segmentation accuracy can affect the final transcription accuracy, we propose to overcome this limitation by using a segmentation-free few-shot detection method, which only requires segmented lines as input.

#### Method description.

Few-shot object detection could be defined as finding one or several instance(s) of an object by providing a cropped example image, called a *support*, in an image that contains the object, called *query* image. Similarly to the previous method, if we have  $N$  classes and we provide  $k$  examples from each symbol class, the problem is considered  $N$ -way  $k$ -shot detection task.

We apply the few-shot transcription method proposed in [173] for recognizing ciphered manuscripts, where the query is the handwritten line image and the supports are one or few examples of each symbol class in the alphabet. The model was trained to match the support set within the query text lines, as shown in Fig. 10.4. The method, trained on known alphabets of symbols, was able to transcribe manuscripts with unseen alphabets. However, the performance was highly enhanced if the model was fine-

tuned (i.e. retrained on some annotated data from the unseen alphabet). For this reason, we use some annotated lines from the runic manuscript to retrain this method.

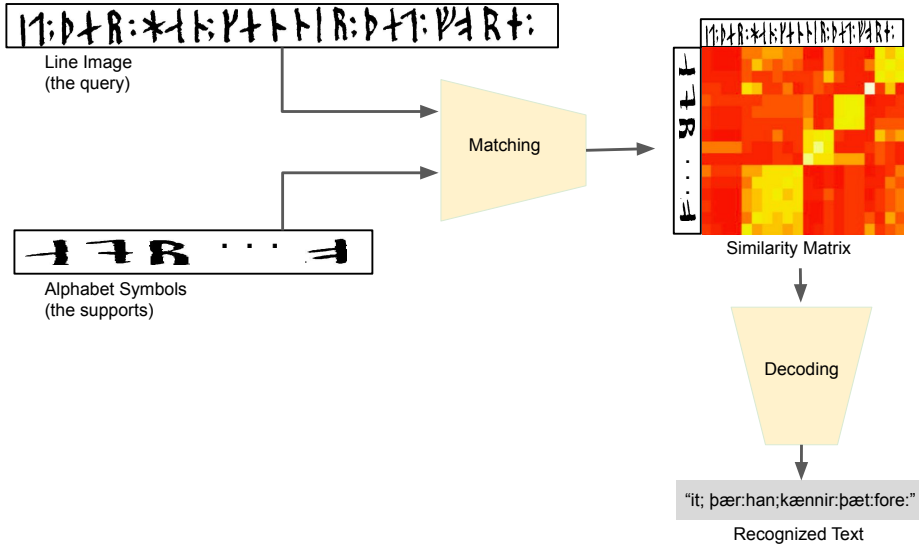


Figure 10.4: Few-shot detection method for transcription: The matching model is applied to construct the similarity matrix between the alphabet and the query line. After that, the matrix is decoded to obtain the final recognized text.

### Adaptation for transcription.

After obtaining the symbols similarities, a decoding algorithm is used to obtain the final transcribed text. From the input image line (the query) and the cipher alphabet images (supports), we obtain the potential bounding boxes coordinates for each one of the support symbols. It may occur that overlapping areas from different supports are detected, each one with the assigned similarity score. So, the final transcription is obtained by selecting the maximum similarities while traversing the matrix from left to right.

## 10.4 Experiments

In this section, we describe the dataset, the metrics and the results obtained in a real use case of a medieval manuscript written in a rare script.



Figure 10.5: Codex Runicus, Arnamagnæan Collection, University of Copenhagen, AM 28 8vo, fol. 86r (<https://www.e-pages.dk/ku/579/>)

#### 10.4.1 Dataset

The *Codex Runicus* [8] is a manuscript consisting of 100 parchment folios (leaves), i.e. 200 pages and is dated to around 1300AD. It contains mainly of legal texts and was produced in a monastic context in the province of Scania in medieval Denmark (now the southernmost part of Sweden). The manuscript text was recorded by mainly one scribe who wrote the pages 1r-82v (pages 1-165) and 84r-91v (pages 168-182) and two more scribes were involved for pages 83r-v (pages 166-167) and 92r-100r (pages 184-200), respectively. An example of this manuscript is shown in Fig. 10.5. The text is written in the Scanian dialect of Old Danish and for the entire manuscript, the runic writing system was used [77, 180].<sup>9</sup>

The legal works contained in the manuscript have been made accessible digitally in

<sup>9</sup>The manuscript is kept at The Arnamagnæan Manuscript Collection at the University of Copenhagen, shelfmark AM 28 8vo. It is also accessible online: <https://www.e-pages.dk/ku/579/>. A description of the manuscript and a bibliography can be found on the Handrit-page <https://handrit.is/da/manuscript/view/AM08-028>.

a project of Det Danske Sprog- og Litteraturselskab.<sup>10</sup> However, this edition provides a so-called normalized language and hence involves an interpretation made by the editor. The original runic text is to date not available as a machine-readable text<sup>11</sup>. Such an edition and transcript is needed for further computational linguistic analysis or questions regarding the writing system.

To evaluate the transcription methods, we took 10 pages the *Codex Runicus* manuscript, written by the same scribe, each one containing 14 lines. Using those pages, we created three different scenarios, progressively increasing the number of training pages. This leads us to the configurations presented in Table 10.2. We start in scenario L1 with 2 pages for training, 2 for validation, and 6 for testing. Then we move each time 2 pages from the testing set to the training set. The purpose of having three different scenarios is to gain information about the influence of the training data size in the different approaches. The aim is to find the method performing best in automatic recognition according to the evaluation metrics while using as few training pages as possible.

Table 10.2: Codex Runicus manuscript pages used in the different experiments scenarios, the numbers are related to the order of the pages in the original manuscript

Scenario	Training pages	Validation pages	Testing pages
L1	124, 109	107, 108	99, 173, 30, 97, 110, 137
L2	124, 109, 110, 137	107, 108	99, 173, 30, 97
L3	124, 109, 110, 137, 30, 97	107, 108	99, 173

### 10.4.2 Evaluation Metrics

As an evaluation metric, we use the Character Error Rate (CER), described in [173]. Formally,  $CER = \frac{S+D+I}{N}$ , where S is the number of substitutions, D of deletions, I of insertions and N is the ground-truth's text length. Thus, the lower the value the better. The CER is shown between 0-100%.

For the unsupervised clustering, we also compute the percentage of missing characters, i.e. the characters that are not transcribed. In these cases, the algorithm assigned too low confidence during classification, so a manual transcription is needed. The confidence is a hyper-parameter set by the user. Obviously, if the confidence threshold increases, the CER decreases, but the percentage of missing symbols also

<sup>10</sup><https://tekstnet.dk/search?search=codex+runicus>.

<sup>11</sup>After having conducted our research on *Codex Runicus*, we heard from a pilot project of a manual XML encoding of the runic manuscript conducted by Paola Peratello at the University of Verona [141]. As part of her current PhD-project, her aim is a full XML encoding of the manuscript on four levels: the facsimile level of the runic script, the diplomatic level with a transliteration into the Latin alphabet, the linguistically normalized level of a common Old Norse language, and linguistic annotation. The encoding of the manuscript will be included in the Medieval Nordic Text Archive ([www.menota.org](http://www.menota.org)). We investigate possible collaborations between our projects.



increases, which means that the user must transcribe more symbols during the validation stage. Thus, the ideal situation is to find the right trade-off between the CER and the percentage of missing symbols.

### 10.4.3 Results

The results obtained for the three scenarios are shown in Table 10.3. We order the methods according to the need of labeled data (from low to high). Note that the CER values between the L1, L2, L3 scenarios are not directly comparable, because the number of test pages differs.

From the results, we observe that the learning-based method (MDLSTM) obtains the worse performance (so, the highest CER) because deep-learning methods suffer

Table 10.3: Obtained results by the transcription methods over the different scenarios. The CER and the Missing Characters are shown in %. Unsupervised clustering methods include with and without User Intervention (UI)

Scenario	Method	Shots	CER	Missing chars
L1	Clustering (without UI)	–	3,2	22,6
	Clustering (with UI)	–	6,7	2,1
	Few-shot Detection	1	7,4	–
		5	6,7	–
	MDLSTM	–	47,9	–
L2	Clustering (without UI)	–	6,7	23,2
	Clustering (with UI)	–	6,9	3
	Few-shot Classification	1	34,2 / Batch	–
		5	5,2	–
	MDLSTM	–	26	–
L3	Clustering (without UI)	–	3,5	22,3
	Clustering (with UI)	–	8,9	3,3
	Few-shot Classification	1	6,5 / Batch	–
		5	8,1	–
	MDLSTM	–	16,9	–

when there is insufficient labeled data to train. As expected, the more training pages (scenarios L2 and L3), the performance improves, but this implies that more transcribed pages must be provided for training.

The unsupervised clustering method without any user intervention obtains the best result (CER of 3,2%) but at the cost of a high percentage of missing characters (over 20%), which implies that the user must transcribe later an important amount of characters (those are marked with the symbol '?'). In case the user cleans the clusters prior to the label propagation step, the missing symbols significantly decrease (2,1%) whereas the CER only increases a bit. Hence, the clustering method with user intervention is preferable.

In the Few-shot classification method, and given the few amount of samples per symbol in the Runic pages, only 1-shot learning is performed. As expected, the best performance was achieved in L3 due to the higher amount of labeled pages, but we have observed that this model has a trend to suffer from overfitting (i.e. learning the appearance of the training symbols by heart, so losing generalization).

In the case of the Few-shot detection method, we observe that the performance is satisfactory in all scenarios, with a CER similar to the clustering method (note that there are no missing characters). As expected, the 5-shot detection setting obtains better performance. We note that for a fair comparison between the scenarios (in time), training them was done for the same number of epochs. Thus, L3 would need more time for training to surpass the other scenarios.

Next, we show some qualitative results in Figure 10.6. The unsupervised clustering method often makes errors when the shape of symbols is similar, for instance between ǫ ("a") and ǫ ("o") or ƿ ("k") and ƿ ("f"). Similarly, the few-shot methods also confuse similar symbols, like the symbol class ":" and the class ";". Concerning the MDLSTM, instead of making errors, it tends to miss some characters. Since the output is a sequence of characters, it tends to skip some frames, mainly because of the few annotated data used for training.

Groundtruth	o l f ; n æ f n ; i : k i r k i u : s o k n ;
Clustering (without UI)	a l f ; n æ k n ; i : k i u k i i : s a k n ;
Clustering (with UI)	o l f ; n æ n ; i : k i r k i u : s a k n ;
MDLSTM	a l m n æ m æ : : i s i u : s t k n ;
Few-shot Clas.	o l f ; n æ f n ; i : k i r k i u : s o k n ;
Few-shot Det.	o l f : n æ f n ; : k i r k i u : s o k n ;

Figure 10.6: Transcription results from scenario L2 of page 173, line 13. The first row corresponds to the ground truth provided by the expert. Errors are shown in orange color, and skipped characters in blue (better viewed in color).

In summary, from the user perspective, we can conclude that the most suitable methods are the few-shot detection and the clustering with user intervention in the L1 and L2 scenarios, since less training data is required.

**10.4.4 Time Needed for Preparing the Training Data**

In this subsection, we focus on the user’s time consumption for preparing the training data, such as transcribing some pages, cleaning clusters, selecting bounding boxes, etc. Table 10.4 shows the time needed for each method in each scenario. For the MDLSTM method, the user must validate and transcribe each segmented line in the pages. For the unsupervised clustering method, the user must provide a transcription label for each cluster. The clustering with user intervention also needs user input for cleaning the clusters. In the few-shot detection method, the user has to validate the segmented lines and provide the bounding boxes of each symbol, which is a rather time-consuming step because it means locating and cropping each symbol in the segmented lines. Besides, in the clustering and few-shot classification methods, the user needs to validate the segmentation of symbols.

Table 10.4: Time consumed for data preparation for training (hours:minutes).

Method	Scenario L1	Scenario L2	Scenario L3
MDLSTM	0:52	1:16	1:51
Clustering (without UI)	0:12	0:16	0:28
Clustering (with UI)	0:32	1:23	2:01
FS-classification	1:18	2:07	3:08
FS-detection	1:16	2:04	3:03

Obviously, scenario L1 requires the lowest preparation time because only 4 pages are labeled (2 for training, 2 for validation). The time consumption raises for L2 and L3 scenarios because the number of training pages increases to 6 and 8 pages, respectively. We observe that there is an important difference between the time needed for each method. For example, the time needed for the clustering without user intervention is very low, whereas it is moderate in the case of MDLSTM, but at the cost of low performance, as shown in Table 10.3. Contrary, the few-shot methods need more time due to the annotation of the bounding boxes for each symbol.

**10.4.5 User Validation**

Next, we analyze the time that the user needs for validating the output of the transcription methods. During this post-processing phase, the user checks the output and corrects the transcription errors. Here, we restrain our analysis to the scenario L2 (4 pages for testing) using the two best-performing methods: clustering with user intervention and few-shot detection.

The validation was performed as follows. The transcription results were provided in txt-documents and were checked next to the digital images of the respective manuscript pages on one desktop. For each page, the specialist counted the number of errors and measured the time needed to check the transcriptions and correct the errors in the txt file. The user validated the transcription results of the two methods on two different days. In general, we noted that on three of the test pages (pages 30, 97 and 99) only very few errors occurred. These errors include mainly confusions of visually similar symbols: ƿ ("e") and ƿ ("æ"), ı ("i") and ı ("t"), ƿ ("f") and ƿ ("k"), and the punctuation marks ":" and ";". The first two symbol pairs and the punctuation marks are – depending on the handwriting and the conservation status of the manuscript page – in many instances also difficult to differentiate for a specialist in runes, and disambiguation is only possible in the later process of the linguistic analysis. On the fourth page (page 173), the number of errors was significantly higher because there is a different handwriting style, with smaller and narrower symbols compared to the other pages. The mistakes produced by the few-shot detection method include again confusion of visually similar symbols, e.g. ƿ ("e") instead of ƿ ("æ"). Moreover, however, slim symbols (such as ";" and ":", as well as ı ("i") and \* ("h")) have been missed out. For the clustering method, there were many missing symbols (indicated by "?") that had to be manually transcribed.

Table 10.5: Number of errors and time needed for validation time vs manual transcription.

Method	Clustering with UI	Few-shot detection	Manual transc.
Page 30	19 errors, 7 min	4 errors, 5 min	21 min
Page 97	16 errors, 7 min	2 errors, 5 min	14 min
Page 99	14 errors, 6 min	2 errors, 5 min	15 min
Page 173	34 errors, 10 min	33 errors, 14 min	14 min
Total	30 min	29 min	64 min

The time needed for the manual validation, as well as the number of mistakes, are presented in Table 10.5. From the user perspective, an interesting point to note is that the validation of the clustering method is quite fast despite a significantly higher amount of mistakes, in comparison with the few-shot detection method. The reason is that the clustering method only transcribes each symbol if its confidence is high, otherwise, it labels it as '?' (missing symbol). Thus, for the user, it is easier and faster to detect and label each '?' symbol rather than searching for errors.

As a reference, we also provide the time needed for a full manual transcription (without automatic tools). The transcription was written into a txt file that was shown together with the digital images of the manuscript pages on one desktop. The transcription included a transliteration of the runes to the roman alphabet in order to facilitate the typing process. After transcribing five pages, the transcriber took a break. In the end, after another break, the first transcription of the ten pages was validated showing the digital pictures and the transcription in the txt-files on one desktop. The

manual transcription of each page took between 14 and 21 minutes. In this time frame, we also include the time for validating the manual transcription. It has to be noted that the transcriber was familiar with the runic writing system and knew the transliteration system by heart. Hence, the manual transcription speed of our specialist has to be regarded as rather high in comparison with a transcriber that meets unfamiliar writing systems.

From Table 10.5, we observe that when the transcription methods obtain a low CER, the validation time is low, making it remarkably faster than manual transcription. For instance, page 30 was manually transcribed in 21 min, which is significantly higher compared to the 5 minutes needed for validating the output of the few-shot detection method, or the 7 minutes of the clustering method. Moreover, when CER increases, like in page 173, the time for validation increases and being closer to the time needed for a full manual transcription. If the amount of errors increases above these values, obviously, the time for correcting errors would explode and surpass the time for a manual transcription.

## **10.5 Recommendations for tools in Digital Humanities**

In the previous section, we tested and evaluated four different transcription methods with regard to their performance and user validation time. Based on the analysis of these results, we here point out the advantages and disadvantages of the different methods from a user perspective, comment on general affordances and constraints of automatic transcription of historical documents written in rare scripts, and raise recommendations for scholars interested in applying HTR tools.

### **10.5.1 Advantages and Disadvantages**

A general observation from the user perspective is that there are many arguments in favor of the use of HTR technology in the transcription of documents in rare scripts – despite the constraints mentioned above. Manual transcription is not only highly time-consuming, but also prone to errors, especially when a transcriber is unfamiliar with a symbol set and/or the single symbols are visually similar and hard to differentiate. It requires utmost concentration and leads to fatigue. Automatic transcription decreases time consumption significantly in the work with larger documents. In addition, and this applies also to the work with shorter documents, manual correction of the transcription output is facilitated since the mistakes or insecurities provoked by automatic transcriptions are systematic. An HTR analysis with a performance of the here-tested few-shot detection method and the unsupervised method gives a transcription for the single symbols and enables to channel the attention of the user to difficult cases. This shortens the time and effort that has to be invested in transcription in general.

However, some general downsides of currently available HTR methods for the philol-

ogist or paleographers have to be mentioned. The algorithms are so far not able to detect and represent philological details such as erasures or other corrections. Different color shades of inks or other material details, such as illustrations in the margin, are also lost in pre-processing. These aspects are important if the purpose of the transcription is an edition of a certain text document. In such cases, HTR technology can only be a help in the work of a transcriber and does not fully replace manual work with historical documents. In other use cases with different purposes and research questions, however, such as for a historical or literary research purpose where the text content is of central significance and the material context of the manuscript is less important, HTR technology speeds up the research process significantly and can replace laborious transcription work.

### 10.5.2 Recommendations

First, we provide recommendations depending on the particularities of the manuscripts to transcribe.

#### **Availability of labeled training data.**

In scenarios where enough annotated training data is available to train, deep-learning-based HTR methods, such as MDLSTM, obtain the best performance. However, in the case of rare scripts, labeled data is barely available, so the MDLSTM has to be discarded. The same occurs with the few-shot classification method, which tends to overfit when few data is available. Hence, in scenarios where labeled data is very limited and/or has to be manually generated, the few-shot detection and clustering methods are preferable. Considering both the transcription errors and the time for validation, the few-shot detection method is the most suitable one.

#### **Multi-writer manuscripts.**

If the manuscript to transcribe contains different handwriting styles, the differences in the shape of symbols will affect all the transcription methods. This means that the user should provide examples for each alphabet symbol for the different handwriting styles so that the system learns those variants. Of course, this implies that the amount of labeled training data has to be increased. If this amount of training data is not likely to be provided, the recommendation is to use the clustering method. The clustering groups samples according to the similarity in shape appearance, hence, it is more likely to group those samples belonging to the same symbol class for each handwriting style. Indeed, clustering could also be useful to visually compare writing systems and differentiate among scribes.

**Cursive Handwriting.**

As we mentioned before, the clustering and few-shot classification methods require the segmentation of the text lines into symbols. In cursive or untidy handwriting, symbols often touch or overlap. This provokes an inaccurate symbol segmentation with the consequent impact in the transcription stage unless the user spends a considerable amount of time in correcting the wrongly segmented symbols. For this reason, MDLSTM and few-shot detection methods are preferable for untidy or cursive handwriting.

**Unknown alphabet.**

Although the few-shot detection method with five shots obtained the best performance in our experiments, such a scenario means that the user has to provide five examples of each symbol in the alphabet. This setting is unproblematic if the writing system is previously known (as is the case for the runes in the here investigated case study) or consists of a small and limited set of distinct symbols. However, for texts with unknown graphic signs and unknown sizes of the symbol set, it might be time-consuming for the user to find five examples of the same symbol. Since this method obtains good performance, it is true that such training time can be later compensated because the user will likely devote fairly little time for post-processing correction. Anyway, it is recommended to use this method only when the manuscript is long and the symbol set of a writing system is fairly transparent for the user. Contrary, when the writing systems are complex, completely unknown, and presumably large, the clustering method is a better choice. First, because it can group symbols that look similar, providing a hint of the underlying alphabet in the manuscript, and second, because it allows for user intervention to clean the clusters. This means that a user can, for instance, automatically transcribe one or two pages, then clean the clusters on the basis of these results and continue with the label propagation and transcription using the cleaned data. Finally, the user has the possibility to tune the confidence threshold for symbol transcription. Hence, a higher threshold value would produce a more accurate transcription (but more "missing characters"), as only the symbols recognized with a high confidence would be transcribed. This scenario, however, means that more time has to be invested in manually transcribing the symbols marked as "?" during the post-processing correction.

**Length of Manuscript.**

Along the experiments, we have observed that all methods need the user to provide labeled data to train, clean clusters, etc. Although this step is only required for the initial setup of the methods, it can limit the suitability of transcription methods for manuscripts with few pages. For example, in the L2 scenario, the total time consumption (including data preparation and user validation) is significantly higher than the

time needed for manually transcribing the four test pages. Indeed, the longer the manuscript, the more suitable is the use of automatic transcription tools (e.g. *Codex Runicus* contains 200 pages). The reason is that the time for data preparation is constant no matter the amount of pages to transcribe, so in longer manuscripts, the training time gets compensated by the higher speed (5-7min/page) for user validation, when compared to a manual transcription (14-15min/page). We have illustrated this evolution in Fig.10.7, where we take the scenario L2 and assume an average time for validation per page. As it can be observed, a manual transcription is faster if the manuscript is shorter than 12 pages, the clustering method is faster beyond that number, whereas the few-shot detection method is preferable when the number of pages to transcribe is above 28.

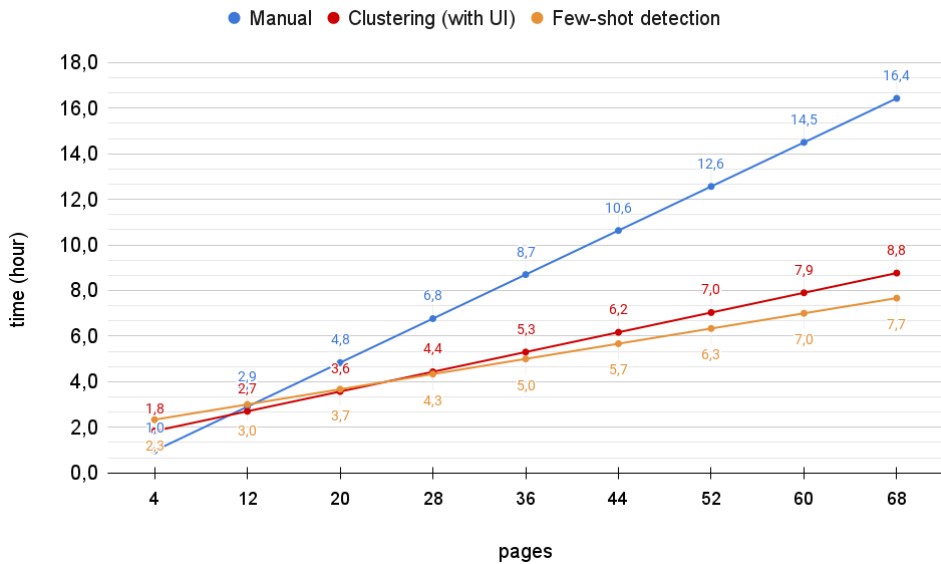


Figure 10.7: Estimated total time needed for transcribing the *Codex Runicus* in scenario L2. We assume that the Few-shot detection method needs 2h for data preparation (6 pages), and 5 min/page for validation; the clustering method with user intervention needs 1'4h for data preparation, and 6'5 min/page for validation; the manual transcription is 14'5 min/page.

### 10.5.3 Summary

In summary, we can conclude that there is no perfect method to be applied to any scenario, so the method to be chosen depends on the particularities of the manuscript to transcribe. We summarize the recommendations for scholars in Table 10.6. In general, the conclusions are:



- Automatic transcription methods are worth it when there are many pages to transcribe, especially if there are few handwriting styles. Otherwise, the amount of labeled data to be trained increases significantly.
- Learning-free methods, like unsupervised clustering, are suitable when symbols are easy to segment and/or the alphabet of symbols is complex or unknown.
- Few-shot detection and MDLSTM methods are preferable for cursive or untidy handwriting styles because they do not need any symbol segmentation step.
- Supervised deep learning-based methods, like MDLSTM, are suitable only when an important amount of labeled data is available to train.

Table 10.6: Summary of recommendations for choosing a transcription method. The symbols mean the following. ✗ means not suitable, = means medium suitability, and ✓ means very suitable.

	Few labeled data	Short texts	Long texts	Complicated or Unknown scripts	Cursive or Untidy text
Deep learning (MDLSTM)	✗	✓	✓	✓	✓
Learning-free (clustering)	✓	=	✓	✓	✗
Few-shot learning (detection)	=	✓	✓	✗	✓
Manual Transcription	✓	✓	✗	=	✗

## 10.6 Conclusion

In this work, we have developed and extensively evaluated the performance of different transcription methods when applied to uncommon scripts. The analysis of the results, both in terms of transcription performance and time consumption for data preparation and post-processing validation has been the basis for providing recommendations from the user perspective. We focus our experiments on the Codex Runicus manuscript, however, we hope that our work can serve as a guide for scholars interested in using transcription tools in the humanities with any script.

Since the best methods for transcribing rare manuscripts are the clustering and few-shot methods, we have started the implementation of both methods as a web service. The clustering method is already freely available <sup>12</sup>, whereas the few-shot detection method will be soon incorporated into our transcription platform.

<sup>12</sup><https://de-crypt.org/service.php>

In the near future, we plan to incorporate interactive tools for easing the validation of the automatically transcribed pages so that the time for post-correction is minimized (for example, highlighting the "missing characters" or suggesting possible transcriptions for ambiguous characters). From the methodological point of view, we will explore incremental learning and relevant feedback, so that our transcription tools can continuously learn from the errors that the user corrects, and hopefully, progressively reduce the transcription errors along the rest of the pages to be transcribed in the manuscript.



# Chapter 11

## Conclusion

*There is no real ending. It's just the place where you stop the story.*  
– Dune, 1965 by Frank Herbert

---

*In this Chapter, we summarize the contributions proposed in this thesis and their application to the historical document analysis problems. We highlight the main success and limitations of our proposed models.*

---

In this thesis, we focused on problems related to handwritten document image enhancement and recognition using different computer vision and machine learning mechanisms. The main challenge to address these problems was the scarcity of data. Thus, we used multiple learning strategies to overcome this issue, e.g. few-shot learning, data generation, self-supervised learning, etc. Moreover, an evaluation from a user perspective was presented to validate the usefulness of the proposed tools to the standard users. So far, the proposed tools and models showed a good performance to facilitate the access to the historical manuscripts (ciphers and handwritten text) by automatizing the enhancement and recognition with a significant gain in terms of time and human labor. In what follows, a brief summary of the contributions.

### 11.1 Summary of the Contributions

With the advent of smart devices, it has become an utmost requirement to ease the digitization and processing of handwritten documents. Historical documents are an important part of the human cultural heritage, thus, our aim was to facilitate the processing of such documents by providing image quality enhancement and handwritten

text recognition methods. This was detailed in Chapter 1 along with the difficulties and challenges that construct the motivation of this thesis.

After that, we present the rest of this thesis in three major parts. The first part focused on the pre-processing of the document image by enhancing its quality. The second part was addressing the recognition of the historical and ciphered handwritten text images with different deep learning models and learning strategies. Finally, the third part was dedicated to evaluating some automatic recognition methods of handwritten text from a user perspective and comparing them with a manual transcription while stating the benefits and drawbacks of each method. In the following, we present the contributions of this thesis in more detail.

- **End-to-end Document Image Enhancement:** A new model for document image enhancement was proposed. At the start, we formulate the problem as an image-to-image translation in an end-to-end fashion, where a conditional GAN was used to tackle it. After that, we propose an improvement of the conditional GAN model by making it focus more on the textual details while enhancing the document image. This was done by adding a mechanism to evaluate the text during training, which consists in a recognizer model. A well-designed loss was used to produce a clean version of the degraded document image while maintaining its readability. Then, we use a transformer model to address the enhancement problem. The new model shows a better performance than the GAN-based models and leads to state-of-the-art performance in the document binarization task.
- **Few-shot Learning for HTR:** To overcome the problem of data scarcity in historical rare documents, like ciphered manuscripts, we used the few-shot learning mechanism as a symbol/object detection problem. Our work was, to the best of our knowledge, the first few-shot learning model for HTR that works at line level. Our proposed model demonstrated that it was possible to accurately recognize a new manuscript with an unseen alphabet by using only 2 labeled pages during training.
- **Progressive Learning for HTR:** Despite the significant gain in data annotation, a human effort is still necessary to annotate a few pages (including transcription and segmentation into bounding boxes). Thus, we proposed to overcome this by a progressive learning technique that used the pseudo-labeling strategy. Our method starts only from a few labeled examples (we used five) from each symbol (or character) belonging to the desired alphabet to recognize. The proposed method was suitable because it led to an important gain in the human labor of annotating historical manuscripts with a minimal loss in terms of accuracy.
- **Data generation for HTR:** Another way to overcome the limitation of data scarcity was to use generative models to produce more data. Given the scarcity constraint, the model should generate data in a few-shot scenario. Thus, we explored the PBL model to generate new examples of any symbol/character belonging to a new alphabet given only a single image. After generating the isolated characters, synthetic lines were created and used to train HTR models. We

showed that this strategy lead to a good performance while requiring only a few annotated data.

- **Self-Supervised Learning for HTR:** When addressing an HTR task, it is easier to obtain unlabeled data than labeled one. In our case, it is possible to benefit from similar unlabeled data, although with a new alphabet, to train a model on. For this reason, we explored the application of self-supervised learning in HTR. We showed that this strategy is useful since a self-supervised pretrained model was leading to a better performance than starting from a scratch model using only labeled data, especially when the amount of the labeled data is reduced.
- **A user perspective evaluation of HTR tools:** As a final contribution to this thesis, we evaluated the HTR models that can be used to transcribe historical manuscripts from the user perspective. We compared the use of different machine / deep learning models as well as manual transcription. From the evaluation, some recommendations for end users when using HTR models were provided.

## 11.2 Discussion

This thesis aimed to contribute to the historical document image analysis and recognition research field. But, we believe that the machine learning strategies and insights proposed within the thesis can also be used to tackle problems in other fields.

During Chapter 3, Chapter ??, and Chapter 5 we showed that our proposed models for document image enhancement can outperform the state-of-the-art approaches on different benchmarks. However, in real scenarios, we found that the performance of these models can decrease if the resolution of the input image is low. Also, the public datasets that we used to evaluate the models were synthetically created. Hence, when enhancing real images the performance can slightly decrease because of the different domains.

The domain gap was also a limitation of our developed models for HTR in low resources. In Chapter 8, the generated lines with BPL were not as realistic as desired. For this reason, we also used training lines generated by data augmentation to reduce the domain gap with real data and slightly improve the performance of the model. However, we believe that the combination of BPL with domain adaptation techniques could lead to further improved results.

The self-supervised approach that was proposed in Chapter 9 was tested on the IAM and CVL datasets for the HTR task. For each experiment, the pretraining and fine-tuning were done on the same dataset. However, it would be interesting to pretrain and fine-tune on data from different domains. Unluckily, obtaining many unlabeled data from the same domain, as required by the SSL models, is difficult in case of the historical and ciphered manuscripts.

To summarize, we can conclude that many improvements can be built on top of

our proposed models within this thesis. Anyway, our models pave the way for more research and improvements to address the recognition of historical manuscripts in low-resource scenarios.

## 11.3 Future Work

Future work could address the limitations of our developed models, stated in the previous section. We list them next:

### 11.3.1 Models Robustness in Document Image Enhancement

As we said before, training on synthetically degraded images can lead to a performance decrease when testing with real data. Also, during the training, the publicly available datasets especially for document binarization are available only in a high resolution. Contrary, in real use-case scenarios, a user might want to enhance an image with low resolution. But, training a supervised approach using *real* and paired data (degraded image with its clean version and GT text) is costly, since it is really hard to obtain this kind of resource. That is why most of the datasets are synthetically made. Thus, one could explore the use of unpaired data. In this way, real images can be easily obtained (degraded images and clean ones). With this data and unpaired training, the model could be trained to produce images that are for the discriminator as real as possible (similar to the clean real documents), while as much readable and real as possible for the recognizer (the recognized text must have a sense).

### 11.3.2 Domain Adaptation

As commented before, in the low-resource scenario, sufficient real data to train models is not available. However, in some cases, similar data (real or synthetically created) could be available. Obviously, when training an HTR model, its performance will decrease when recognizing images from a domain different from the training one. This phenomenon was faced in this thesis when designing the few-shot model (fine tuning on the low resource data was necessary to improve the results) or when generating the synthetic data (we had to combine data augmentation and data generation to reduce the domain gap). To overcome the domain shift issue, domain adaptation [190] techniques could be explored. With domain adaptation, the performance of our models can be further improved since it will allow us to enrich the training by using similar handwritten text images.

### 11.3.3 Continual Learning

The historical and ciphered manuscripts addressed in this thesis are written using different alphabets (including different languages). As a consequence, different models were developed, since each one was trained for a specific alphabet. However, some useful information could be wasted using this strategy. Thus, one interesting idea to explore is to benefit from the previously learned weights, which means to *continue* learning other alphabets while avoiding catastrophic forgetting. Continual learning [100] aims to develop the mechanisms to avoid the catastrophic forgetting issue. By employing continual learning techniques, we believe that we could create better and more generic approaches. Also, it could facilitate the update of existing models by adding the ability to recognize new languages or alphabets.





# List of Contributions

*Coming together is a beginning; keeping together is progress;  
working together is success.*  
by Henry Ford

## Topics

The main topic of this dissertation is the development of better document image enhancement and recognition systems. However, this thesis has also generated other side contributions on other topics that had raised our attention in the same field.

- **Document Image enhancement:** This topic investigates the problem of enhancing the quality of document images (binarization, cleaning, watermark removal, etc).
- **Document Image Recognition:** This task refers to algorithms and techniques that are applied to images of documents to obtain a computer-readable description from pixel data. In particular historical and handwritten documents with rare manuscripts, N-gram spotting, alphabet matching, image text alignment, etc.
- **Transcription Tools/Methods Development and Evaluation:** This task refers to the development of tools for automatic transcription and the evaluation of different transcription methods efficiency from a user perspective.

## International Journals

1. **Souibgui, Mohamed Ali,** and Yousri Kessentini. "DE-GAN: a conditional generative adversarial network for document enhancement." in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

2. Jemni, Sana Khamekhem, **Mohamed Ali Souibgui**, Yousri Kessentini, and Alicia Fornés. "Enhance to read better: a multi-task adversarial network for handwritten document image enhancement." in *Pattern Recognition*, 123 (2022).
3. **Souibgui, Mohamed Ali**, Alicia Fornés, Yousri Kessentini, and Beáta Megyesi. "Few shots are all you need: A progressive learning approach for low resource handwritten text recognition." in *Pattern Recognition Letters*, 160 (2022).
4. **Souibgui, Mohamed Ali**, Asma Bensalah, Jialuo Chen, Alicia Fornés, and Michelle Waldispühl. "A User Perspective on HTR methods for the Automatic Transcription of Rare Scripts: The Case of Codex Runicus." in *Journal on Computing and Cultural Heritage (JOCCH)* (2022).

## International Conferences

1. Chen, Jialuo, **Mohamed Ali Souibgui**, Alicia Fornés, and Beáta Megyesi. "A Web-based Interactive Transcription Tool for Encrypted Manuscripts." In *International Conference on Historical Cryptology 2020*.
2. **Souibgui, Mohamed Ali**, Yousri Kessentini, and Alicia Fornés. "A conditional gan based approach for distorted camera captured documents recovery." In *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, pp. 215-228. Springer, Cham, 2020.
3. **Souibgui, Mohamed Ali**, Alicia Fornés, Yousri Kessentini, and Crina Tudor. "A few-shot learning approach for historical ciphered manuscript recognition." In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5413-5420. IEEE, 2021.
4. Chen, Jialuo, **Mohamed Ali Souibgui**, Alicia Fornés, and Beáta Megyesi. "Un-supervised Alphabet Matching in Historical Encrypted Manuscript Images." In *International Conference on Historical Cryptology*, pp. 34-37. 2021.
5. **Souibgui, Mohamed Ali**, Ali Furkan Biten, Sounak Dey, Alicia Fornés, Yousri Kessentini, Lluís Gomez, Dimosthenis Karatzas, and Josep Lladós. "One-shot compositional data generation for low resource handwritten text recognition." In *the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 935-943. 2022.
6. **Souibgui, Mohamed Ali**, Sanket Biswas, Sana Khamekhem Jemni, Yousri Kessentini, Alicia Fornés, Josep Lladós, and Umapada Pal. "Docentr: An end-to-end document image enhancement transformer." *2022 26th International Conference on Pattern Recognition (ICPR)* (2022).
7. Magnifico, Giacomo, Beáta Megyesi, **Mohamed Ali Souibgui**, Jialuo Chen, and Alicia Fornés. "Lost in Transcription of Graphic Signs in Ciphers." In *International Conference on Historical Cryptology*, pp. 153-158. 2022.

8. De Gregorio, Giuseppe, Sanket Biswas, **Mohamed Ali Souibgui**, Asma Bensalah, Josep Lladós, Alicia Fornés, and Angelo Marcelli. "A Few Shot Multi-Representation Approach for N-gram Spotting in Historical Manuscripts" In *International Conference on Frontiers in Handwriting Recognition*. 2022.
9. **Souibgui, Mohamed Ali**, Sanket Biswas, Andres Mafla, Ali Furkan Biten, Alicia Fornés, Yousri Kessentini, Josep Lladós, Lluís Gomez, and Dimosthenis Karatzas. "Text-DIAE: A Self-Supervised Degradation Invariant Autoencoders for Text Recognition and Document Enhancement." In *2023 AAAI Conference on Artificial Intelligence (AAAI)*. (Under Review)

## International Workshops

1. Torras, Pau, **Mohamed Ali Souibgui**, Jialuo Chen, and Alicia Fornés. "A Transcription Is All You Need: Learning to Align Through Attention." In *International Conference on Document Analysis and Recognition Workshops*, pp. 141-146. Springer, Cham, 2021.

## GitHub Repositories

1. <https://github.com/dali92002/DE-GAN>
2. <https://github.com/dali92002/DocEnTR>
3. <https://github.com/dali92002/HTRbyMatching>
4. <https://github.com/dali92002/SSL-OCR>

## Awards

1. **ICPR2020 Best Student Paper Award:** Given by the ICPR2020 organizing committee for the paper entitled "A Few-shot Learning Approach for Historical Ciphered Manuscript Recognition in the track of Document and Media Analysis."



# Bibliography

- [1] A. F. Bickley. Bickley diary dataset. *Private Journal (unpublished), donated to the Methodist Church Archives, Singapore by Erin Bickley*, 1926.
- [2] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anshel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15302–15312, 2021.
- [3] Muhammad Zeshan Afzal, Joan Pastor-Pellicer, Faisal Shafait, Thomas M Breuel, Andreas Dengel, and Marcus Liwicki. Document image binarization using lstm: A sequence learning approach. In *Proceedings of the 3rd international workshop on historical document imaging and processing*, pages 79–84, 2015.
- [4] Younes Akbari, Somaya Al-Maadeed, and Kalthoum Adam. Binarization of degraded document images using convolutional neural networks and wavelet-based multichannel images. *IEEE Access*, 8:153517–153534, 2020.
- [5] Mohsen Annabestani and Mahdi Saadatmand-Tarzjan. A new threshold selection method based on fuzzy expert systems for separating text from the background of document images. *Iranian journal of science and technology, transactions of electrical engineering*, 43(1):219–231, 2019.
- [6] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. *ICCV*, 2021.
- [7] Kohei Arai and Ali Ridho Barakbah. Hierarchical k-means: an algorithm for centroids initialization for k-means. *Reports of the Faculty of Science and Engineering*, 36(1):25–31, 2007.
- [8] Codex Runicus. <https://www.e-pages.dk/ku/579/>.
- [9] Seyed Morteza Ayatollahi and Hossein Ziaei Nafchi. Persian heritage image binarization competition (phibc 2012). In *2013 First Iranian Conference on Pattern Recognition and Image Analysis (PRIA)*, pages 1–4. IEEE, 2013.

- 
- [10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [12] Steve Bako, Soheil Darabi, Eli Shechtman, Jue Wang, Kalyan Sunkavalli, and Pradeep Sen. Removing shadows from images of documents. In *Asian Conference on Computer Vision*, pages 173–183. Springer, 2016.
- [13] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.
- [14] Arnau Baró, Jialuo Chen, Alicia Fornés, and Beáta Megyesi. Towards a generic unsupervised method for transcription of encoded manuscripts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, pages 73–78, 2019.
- [15] Sergey Bartunov and Dmitry Vetrov. Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pages 670–678. PMLR, 2018.
- [16] Asma Bensalah, Pau Riba, Alicia Fornés, and Josep Lladós. Shoot less and sketch more: an efficient sketch classification via joining graph neural networks and few-shot learning. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 1, pages 80–85. IEEE, 2019.
- [17] Pavel Vladimirovich Bezmaternykh, Dmitrii Alexeevich Ilin, and Dmitry Petrovich Nikolaev. U-net-bin: hacking the document image binarization contest. 2019.
- [18] Ujjwal Bhattacharya, Réjean Plamondon, Souvik Dutta Chowdhury, Pankaj Goyal, and Swapan K Parui. A sigma-lognormal model-based approach to generating large synthetic online handwriting sample databases. *IJDAR*, 20(3):155–171, 2017.
- [19] Ankan Kumar Bhunia, Ayan Kumar Bhunia, Prithaj Banerjee, Aishik Konwer, Abir Bhowmick, Partha Pratim Roy, and Ummapada Pal. Word level font-to-font image translation using convolutional recurrent generative adversarial networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3645–3650. IEEE, 2018.
- [20] Ankan Kumar Bhunia, Ayan Kumar Bhunia, Aneeshan Sain, and Partha Pratim Roy. Improving document binarization via adversarial noise-texture augmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2721–2725. IEEE, 2019.

- [21] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5672–5681, 2021.
- [22] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. *arXiv preprint arXiv:2112.12494*, 2021.
- [23] Théodore Bluche. *Deep neural networks for large vocabulary handwritten text recognition*. PhD thesis, Paris 11, 2015.
- [24] Théodore Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in neural information processing systems*, 29, 2016.
- [25] Bartosz Bogacz, Nicholas Howe, and Hubert Mara. Segmentation free spotting of cuneiform using part structured models. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 301–306. IEEE, 2016.
- [26] Jean-Christophe Burie, Mickaël Coustaty, Setiawan Hadi, Made Windu Antara Kesiman, Jean-Marc Ogier, Erick Paulus, Kimheng Sok, I Made Gede Sunarya, and Dona Valy. Icfhr2016 competition on the analysis of handwritten text in images of balinese palm leaf manuscripts. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 596–601. IEEE, 2016.
- [27] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. A selectional auto-encoder approach for document image binarization. *Pattern Recognition*, 86:37–47, 2019.
- [28] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.
- [29] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.
- [30] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.
- [31] Junbum Cha, Sanghyuk Chun, Gayoung Lee, Bado Lee, Seonghyeon Kim, and Hwalsuk Lee. Few-shot compositional font generation with dual memory. *arXiv preprint arXiv:2005.10510*, 2020.
- [32] Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. Generating handwritten chinese characters using cyclegan. In *2018 WACV*, pages 199–207. IEEE, 2018.



- [33] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.
- [34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [35] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.
- [36] Xiaogang Chen, Xiangjian He, Jie Yang, and Qiang Wu. An effective document image deblurring algorithm. In *CVPR 2011*, pages 369–376. IEEE, 2011.
- [37] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.
- [38] Danni Cheng, Xiang Li, Wei-Hong Li, Chan Lu, Fake Li, Hua Zhao, and Wei-Shi Zheng. Large-scale visible watermark detection and removal with deep convolutional networks. In *Chinese conference on pattern recognition and computer vision (prcv)*, pages 27–40. Springer, 2018.
- [39] Mohamed Cheriet, Joseph N Said, and Ching Y Suen. A recursive thresholding technique for image segmentation. *IEEE transactions on image processing*, 7(6):918–921, 1998.
- [40] Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. In *British Machine Vision Conference (BMVC)*. Springer, 2019.
- [41] Chien-Hsing Chou, Wen-Hsiung Lin, and Fu Chang. A binarization method with learning-built rules for document images produced by cameras. *Pattern Recognition*, 43(4):1518–1530, 2010.
- [42] Arindam Chowdhury and Lovekesh Vig. An efficient end-to-end neural model for handwritten text recognition. *arXiv preprint arXiv:1807.07965*, 2018.
- [43] Garima Chutani, Tushar Patnaik, and Vimal Dwivedi. An improved approach for automatic denoising and binarization of degraded document images based on region localization. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2272–2278. IEEE, 2015.
- [44] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile. *arXiv preprint arXiv:1901.02199*, 2019.

- [45] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021.
- [46] Quang-Vinh Dang and Guee-Sang Lee. Document image binarization with stroke boundary feature guided network. *IEEE Access*, 9:36924–36936, 2021.
- [47] Valentin De Bortoli, Agnès Desolneux, Bruno Galerne, and Arthur Leclaire. Patch redundancy in images: A statistical testing framework and some applications. *SIAM Journal on Imaging Sciences*, 12(2):893–926, 2019.
- [48] Stanislas Dehaene. *Consciousness and the brain: Deciphering how the brain codes our thoughts*. Penguin, 2014.
- [49] Tali Dekel, Michael Rubinstein, Ce Liu, and William T Freeman. On the effectiveness of visible watermarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2146–2154, 2017.
- [50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [51] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.
- [52] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [53] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.
- [54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [56] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

- [57] Hao Feng, Yuechen Wang, Wengang Zhou, Jiajun Deng, and Houqiang Li. Doctr: Document image transformer for geometric unwarping and illumination correction. *arXiv preprint arXiv:2110.12942*, 2021.
- [58] Steven Roger Fischer. *History of writing*. Reaktion books, 2003.
- [59] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roei Litman. Scrabblegan: Semi-supervised varying length handwritten text generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4324–4333, 2020.
- [60] Alicia Fornés, Beáta Megyesi, and Joan Mas. Transcription of encoded manuscripts with image processing techniques. In *DH*, 2017.
- [61] Volkmar Frinken, Markus Baumgartner, Andreas Fischer, and Horst Bunke. Semi-supervised learning for cursive handwriting recognition using keyword spotting. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 49–54. IEEE, 2012.
- [62] Volkmar Frinken and Horst Bunke. Continuous handwritten script recognition. In *Handbook of Document Image Processing and Recognition*, pages 391–425. Springer, 2014.
- [63] Basilios Gatos, Ioannis Pratikakis, and Stavros J Perantonis. An adaptive binarization technique for low quality historical documents. In *International Workshop on Document Analysis Systems*, pages 102–113. Springer, 2004.
- [64] Basilios Gatos, Konstantinos Ntirogiannis, and Ioannis Pratikakis. Icdar 2009 document image binarization contest (dibco 2009). In *2009 10th International conference on document analysis and recognition*, pages 1375–1382. IEEE, 2009.
- [65] Arna Ghosh, Biswarup Bhattacharya, and Somnath Basu Roy Chowdhury. Handwriting profiling using generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [66] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.
- [67] Raul Gomez, Ali Furkan Biten, Lluís Gomez, Jaume Gibert, Dimosthenis Karatzas, and Marçal Rusiñol. Selective style transfer for text. In *2019 ICDAR*, pages 805–812. IEEE, 2019.
- [68] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [69] Alex Graves. Offline arabic handwriting recognition with multidimensional recurrent neural networks. In *Guide to OCR for Arabic scripts*, pages 297–313. Springer, 2012.

- [70] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
- [71] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.
- [72] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in neural information processing systems*, 21, 2008.
- [73] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.
- [74] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [75] He, Sheng and Schomaker, Lambert. Deepotsu: Document enhancement and binarization using iterative deep learning. *Pattern Recognition*, 91:379–390, 2019.
- [76] Rachid Hedjam, Mohamed Cheriet, and Margaret Kalacska. Constrained energy maximization and self-referencing method for invisible ink detection from multispectral historical document images. In *2014 22nd International Conference on Pattern Recognition*, pages 3026–3031. IEEE, 2014.
- [77] Ake Holmback and Elias Wessén. *Svenska landskapslagar. Serie 4 Skanelagen och Gutalagen*, volume 4. Geber, Stockholm, 1943.
- [78] RJ Howard, MJ Brammer, A David, P Woodruff, S Williams, et al. The anatomy of conscious vision: an fmri study of visual hallucinations. *Nature neuroscience*, 1(8):738–742, 1998.
- [79] Nicholas R Howe. Document binarization with automatic parameter tuning. *International journal on document analysis and recognition (ijdar)*, 16(3):247–258, 2013.
- [80] Michal Hradiš, Jan Kotera, Pavel Zemčík, and Filip Šroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, volume 10, 2015.
- [81] I. Pratikakis, B. Gatos, K. Ntirogiannis. H-dibco 2010 - handwritten document image binarization competition. In *International Conference on Frontiers in Handwriting Recognition*, pages 727—732. IEEE, 2010.

- [82] I. Pratikakis, B. Gatos, K. Ntirogiannis. Icdar 2011 document image binarization contest (dibco 2011). In *2011 International Conference on Document Analysis and Recognition*, page 1506–1510, 2011.
- [83] I. Pratikakis, K. Zagoris, G. Barlas, B. Gatos. ICFHR 2016 handwritten document image binarization contest (H-DIBCO 2016). In *2016 International Conference on Frontiers in Handwriting Recognition*, pages 619—623. IEEE, 2016.
- [84] I. Pratikakis, K. Zagoris, G. Barlas, B. Gatos. Icdar 2017 competition on document image binarization (dibco 2017). In *2017 International Conference on Document Analysis and Recognition*, pages 1395—1403. IEEE, 2017.
- [85] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [86] J. Guo, C. He, X. Zhang. Nonlinear edge-preserving diffusion with adaptive source for document images binarization. *Appl. Math. and Comput.*, 351:8–22, 2019.
- [87] J. Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 67—72. IEEE, 2017.
- [88] Martin Szummer Tommi Jaakkola and Martin Szummer. Partially labeled classification with markov random walks. *Advances in neural information processing systems (NIPS)*, 14:945–952, 2002.
- [89] Sana Khamekhem Jemni, Yousri Kessentini, and Slim Kanoun. Improving recurrent neural networks for offline arabic handwriting recognition by combining different language models. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(12):2052007, 2020.
- [90] Sana Khamekhem Jemni, Yousri Kessentini, Slim Kanoun, and Jean-Marc Ogier. Offline arabic handwriting recognition using blstms combination. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 31–36. IEEE, 2018.
- [91] Sana Khamekhem Jemni, Mohamed Ali Souibgui, Yousri Kessentini, and Alicia Fornés. Enhance to read better: A multi-task adversarial network for handwritten document image enhancement. *Pattern Recognition*, 123:108370, 2022.
- [92] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.
- [93] Lei Kang, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusiñol. Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. *Pattern Recognition*, 112:107790, 2021.

- [94] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Ganwriting: Content-conditioned generation of styled handwritten word images. In *ECCV*, pages 273–289. Springer, 2020.
- [95] Lei Kang, J Ignacio Toledo, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusinol. Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In *German Conference on Pattern Recognition*, pages 459–472. Springer, 2018.
- [96] Seokjun Kang, Brian Kenji Iwana, and Seiichi Uchida. Complex image processing with less data document image binarization by integrating multiple pretrained u-net modules. *Pattern Recognition*, 109:107577, 2021.
- [97] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [98] Benjamin Kiessling, Robin Tissot, Peter Stokes, and Daniel Stökl Ben Ezra. escriptorium: An open source platform for historical document analysis. In *2019 International Conference on Document Analysis and Recognition Workshops (IC-DARW)*, volume 2, pages 19–19. IEEE, 2019.
- [99] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [100] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [101] Netanel Kligler, Sagi Katz, and Ayellet Tal. Document enhancement using visibility detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2374–2382, 2018.
- [102] Kevin Knight, Beáta Megyesi, and Christiane Schaefer. The copiale cipher. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 2–9, 2011.
- [103] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1920–1929, 2019.
- [104] Aishik Konwer, Ayan Kumar Bhunia, Abir Bhowmick, Ankan Kumar Bhunia, Prithaj Banerjee, Partha Pratim Roy, and Umapada Pal. Staff line removal using generative adversarial networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1103–1108. IEEE, 2018.

- [105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [106] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [107] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [108] 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.
- [109] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [110] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.
- [111] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10649–10657, 2019.
- [112] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2927–2936, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- [113] Gundram Leifert, Roger Labahn, and Joan Andreu Sánchez. Two semi-supervised training approaches for automated text recognition. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 145–150. IEEE, 2020.
- [114] Thibault Lelore and Frederic Bouchara. Fair: a fast algorithm for document image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):2039–2048, 2013.
- [115] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

- [116] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. pages 1–10, 06 2019.
- [117] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.
- [118] Dong Liang, Ling Li, Mingqiang Wei, Shuo Yang, Liyan Zhang, Wenhan Yang, Yun Du, and Huiyu Zhou. Semantically contrastive learning for low-light image enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1555–1563, 2022.
- [119] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.
- [120] Hao Liu, Bin Wang, Zhimin Bao, Mobai Xue, Sheng Kang, Deqiang Jiang, Yinsong Liu, and Bo Ren. Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2022.
- [121] Francesco Lombardi and Simone Marinai. Deep learning for historical document analysis and recognition—a survey. *Journal of Imaging*, 6(10):110, 2020.
- [122] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [123] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017.
- [124] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.
- [125] Sabri A Mahmoud, Irfan Ahmad, Wasfi G Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A Fink. Khatt: An open arabic offline handwritten text database. *Pattern Recognition*, 47(3):1096–1112.
- [126] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016.
- [127] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.



- [128] Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. The decode database: Collection of historical ciphers and keys. In *The 2nd International Conference on Historical Cryptology, HistoCrypt 2019, June 23-26 2019, Mons, Belgium*, pages 69–78, 2019.
- [129] Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldspühl. Decryption of historical manuscripts: the decrypt project. *Cryptologia*, 44(6):545–559, 2020.
- [130] Sergey Milyaev, Olga Barinova, Tatiana Novikova, Pushmeet Kohli, and Victor Lempitsky. Fast and accurate scene text understanding with image binarization and off-the-shelf ocr. *International Journal on Document Analysis and Recognition (IJ DAR)*, 18(2):169–182, 2015.
- [131] Reza Farrahi Moghaddam and Mohamed Cheriet. A variational approach to degraded document enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1347–1361, 2009.
- [132] Guenter Muehlberger, Louise Seaward, Melissa Terras, Sofia Ares Oliveira, Vicente Bosch, Maximilian Bryan, Sebastian Colutto, Hervé Déjean, Markus Diem, Stefan Fiel, Basilis Gatos, Albert Greinöcker, Tobias Grüning, Guenter Hackl, Vili Haukkovaara, Gerhard Heyer, Lauri Hirvonen, Tobias Hodel, Matti Jokin, Philip Kahle, Mario Kallio, Frederic Kaplan, Florian Kleber, Roger Labahn, Maria Lang Eva, Sören Laube, Gundram Leifert, Georgios Louloudis, Rory McNicholl, Jean-Luc Meunier, Johannes Michael, Elena Mühlbauer, Nathanael Philipp, Ioannis Pratikakis, Joan Puigcerver Pérez, Hannelore Putz, George Retsinas, Verónica Romero, Robert Sablatnig, Andreu Sánchez Joan, Philip Schofield, Giorgos Sfikas, Christian Sieber, Nikolaos Stamatopoulos, Tobias Strauß, Tamara Terbul, Héctor Toselli Alejandro, Berthold Ulreich, Mauricio Villegas, Enrique Vidal, Johanna Walcher, Max Weidemann, Herbert Wurster, and Konstantinos Zagoris. Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. *Journal of Documentation*, 75(5):954–976, 2019.
- [133] Akihiro Nakamura and Tatsuya Harada. Revisiting fine-tuning for few-shot learning. *arXiv preprint arXiv:1910.00216*, 2019.
- [134] Arthur F. S. Neto, Byron L. D. Bezerra, Alejandro H. Toselli, and Estanislau B. Lima. HTR-Flor: a deep learning system for offline handwritten text recognition. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 54–61, 2020.
- [135] Wayne Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, 1985.
- [136] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

- [137] Konstantinos Ntirogiannis, Basilis Gatos, and Ioannis Pratikakis. Icfhr2014 competition on handwritten document image binarization (h-dibco 2014). In *2014 14th International conference on frontiers in handwriting recognition*, pages 809–813. IEEE, 2014.
- [138] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [139] Ram Krishna Pandey and AG Ramakrishnan. Language independent single document image super-resolution using cnn for improved recognition. *arXiv preprint arXiv:1701.08835*, 2017.
- [140] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.
- [141] Paola Peratello. Codex Runicus (AM 28 8vo): A pilot project for encoding a runic manuscript. *Umanistica Digitale*, 9:155–169, Dec. 2020.
- [142] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M. Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [143] Neerad Phansalkar, Sumit More, Ashish Sabale, and Madhuri Joshi. Adaptive local thresholding for detection of nuclei in diversity stained cytology images. In *2011 International conference on communications and signal processing*, pages 218–220. IEEE, 2011.
- [144] Réjean Plamondon and Moussa Djioua. A multi-level representation paradigm for handwriting stroke generation. *Human movement science*, 25(4-5):586–607, 2006.
- [145] Réjean Plamondon and Wacef Guerfali. The generation of handwriting with delta-lognormal synergies. *Biological Cybernetics*, 78(2):119–132, 1998.
- [146] I. Pratikakis, K. Zagori, P. Kaddas, and B. Gatos. Icfhr 2018 competition on handwritten document image binarization (h-dibco 2018). In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 489–493, 2018.
- [147] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. Icdar 2011 document image binarization contest (dibco 2011). In *2011 International Conference on Document Analysis and Recognition*, pages 1506–1510, 2011.
- [148] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. Icfhr 2012 competition on handwritten document image binarization (h-dibco 2012). In *2012 international conference on frontiers in handwriting recognition*, pages 817–822. IEEE, 2012.

- [149] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. Icdar 2013 document image binarization contest (dibco 2013). In *2013 12th International Conference on Document Analysis and Recognition*, pages 1471–1476. IEEE, 2013.
- [150] Ioannis Pratikakis, Konstantinos Zagori, Panagiotis Kaddas, and Basilis Gatos. Icfhr 2018 competition on handwritten document image binarization (h-dibco 2018). In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 489–493, 2018.
- [151] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. Icfhr2016 handwritten document image binarization contest (h-dibco 2016). In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 619–623, 2016.
- [152] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. Icdar2017 competition on document image binarization (dibco 2017). In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1395–1403. IEEE, 2017.
- [153] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 67–72. IEEE, 2017.
- [154] Q.N. Vo, S.H. Kim, H.J. Yang, G. Lee. Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, 74:568—586, 2018.
- [155] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*, 2015.
- [156] R.D. Lins. Nabuco—Two Decades of Processing Historical Documents in Latin America. *J. Univers. Comput. Sci.*, 17:151–161, 2011.
- [157] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [158] Leonard Rothacker, Denis Fisseler, Gerfrid GW Müller, Frank Weichert, and Gernot A Fink. Retrieving cuneiform structures in a segmentation-free word spotting framework. In *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*, pages 129–136, 2015.
- [159] Ahmed Cheikh Rouhou, Marwa Dhiaf, Yousri Kessentini, and Sinda Ben Salem. Transformer-based approach for joint handwriting and named entity recognition in historical document. *Pattern Recognition Letters*, 2021.

- [160] S. K. Bera, S. Ghosh, S. Bhowmik, et al. A non-parametric binarization method based on ensemble of clustering algorithms. *Multimedia Tools and Applications*, 80:7653–7673, 2021.
- [161] Adolfo Santoro and Angelo Marcelli. A novel procedure to speed up the transcription of historical handwritten documents by interleaving keyword spotting and user validation. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1226–1230. IEEE, 2019.
- [162] Adolfo Santoro and Angelo Marcelli. Using keyword spotting systems as tools for the transcription of historical handwritten documents: Models and procedures for performance evaluation. *Pattern Recognition Letters*, 131:329–335, 2020.
- [163] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, Apr 2018.
- [164] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000.
- [165] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1162–1167. IEEE, 2017.
- [166] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 2051–2058. IEEE, 2001.
- [167] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
- [168] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.
- [169] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
- [170] Mohamed Ali Souibgui, Sanket Biswas, Sana Khamekhem Jemni, Yousri Kessentini, Alicia Fornés, Josep Lladós, and Umapada Pal. Docentr: An end-to-end document image enhancement transformer. *arXiv preprint arXiv:2201.10252*, 2022.
- [171] Mohamed Ali Souibgui, Ali Furkan Biten, Sounak Dey, Alicia Fornés, Yousri Kessentini, Lluís Gomez, Dimosthenis Karatzas, and Josep Lladós. One-shot compositional data generation for low resource handwritten text recognition. *arXiv preprint arXiv:2105.05300*, 2021.

- [172] Mohamed Ali Souibgui, Ali Furkan Biten, Sounak Dey, Alicia Fornés, Yousri Kessentini, Lluís Gomez, Dimosthenis Karatzas, and Josep Lladós. One-shot compositional data generation for low resource handwritten text recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 935–943, 2022.
- [173] Mohamed Ali Souibgui, Alicia Fornés, Yousri Kessentini, and Crina Tudor. A few-shot learning approach for historical ciphered manuscript recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5413–5420. IEEE, 2021.
- [174] Mohamed Ali Souibgui and Yousri Kessentini. De-gan: A conditional generative adversarial network for document enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [175] Mohamed Ali Souibgui, Yousri Kessentini, and Alicia Fornés. A conditional gan based approach for distorted camera captured documents recovery. In *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*. Springer, 2020.
- [176] Nikita Srivatsan, Jonathan T Barron, Dan Klein, and Taylor Berg-Kirkpatrick. A deep factorization of style and structure in fonts. *arXiv preprint arXiv:1910.00748*, 2019.
- [177] Bolan Su, Shijian Lu, and Chew Lim Tan. Robust document image binarization technique for degraded document images. *IEEE transactions on image processing*, 22(4):1408–1417, 2012.
- [178] Jorge Sueiras, Victoria Ruiz, Angel Sanchez, and Jose F Velez. Offline continuous handwriting recognition using sequence to sequence neural networks. *Neuro-computing*, 289:119–128, 2018.
- [179] T. Bluche and R. Messina. Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition. In *Proceeding of International Conference on Document Analysis and Recognition (ICDAR)*, pages 646–651. IEEE, 2017.
- [180] Ditlev Tamm and Helle Vogt. *The Danish medieval laws. The laws of Scania, Zealand and Jutland*. Medieval Nordic Laws. Routledge, London, New York, 2016.
- [181] Milad Omrani Tamrin, Mohammed El-Amine Ech-Cherif, and Mohamed Cheriet. A two-stage unsupervised deep learning framework for degradation removal in ancient documents. In *Pattern Recognition. ICPR International Workshops and Challenges*, pages 292–303. Springer International Publishing, 2021.
- [182] Chew Lim Tan, Li Zhang, Zheng Zhang, and Tao Xia. Restoring warped document images through 3d shape modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):195–208, 2005.

- [183] Chris Tensmeyer and Tony Martinez. Document image binarization with fully convolutional neural networks. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 99–104. IEEE, 2017.
- [184] Yao-Hung Hubert Tsai and Ruslan Salakhutdinov. Improving one-shot learning through fusing side information. *arXiv preprint arXiv:1710.08347*, 2017.
- [185] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [186] 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*, pages 1096–1103, 2008.
- [187] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
- [188] Quang Nhat Vo, Soo Hyung Kim, Hyung Jeong Yang, and Guesang Lee. Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, 74:568–586, 2018.
- [189] B. Wang and C. L. P. Chen. An effective background estimation method for shadows removal of document images. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3611–3615, 2019.
- [190] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.
- [191] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.
- [192] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.
- [193] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural networks: Tricks of the trade*, pages 639–655. Springer, 2012.
- [194] Jinlin Wu, Hailin Shi, Shu Zhang, Zhen Lei, Yang Yang, and Stan Z Li. De-mark gan: Removing dense watermark with generative adversarial network. In *2018 International Conference on Biometrics (ICB)*, pages 69–74. IEEE, 2018.

- [195] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wei Bian, and Yi Yang. Progressive learning for person re-identification with one example. *IEEE Transactions on Image Processing*, 28(6):2872–2881, 2019.
- [196] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25, 2012.
- [197] Wei Xiong, Xiuhong Jia, Jingjing Xu, Zijie Xiong, Min Liu, and Juan Wang. Historical document image binarization using background estimation and energy minimization. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3716–3721. IEEE, 2018.
- [198] Wei Xiong, Jingjing Xu, Zijie Xiong, Juan Wang, and Min Liu. Degraded historical document image binarization using local features and support vector machine (svm). *Optik*, 164:218–223, 2018.
- [199] Chaoran Xu, Yao Lu, and Yuanpin Zhou. An automatic visible watermark removal technique using image inpainting algorithms. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 1152–1157. IEEE, 2017.
- [200] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multimodal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.
- [201] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.
- [202] Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. Decipherment of historical manuscript images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 78–85. IEEE, 2019.
- [203] Francisco Zamora-Martínez, S España-Boquera, and MJ Castro-Bleda. Behaviour-based clustering of neural networks applied to document enhancement. In *International Work-Conference on Artificial Neural Networks*, pages 144–151. Springer, 2007.
- [204] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.
- [205] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Adaptive text recognition through visual matching. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.
- [206] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

- [207] Xinyun Zhang, Binwu Zhu, Xufeng Yao, Qi Sun, Ruiyu Li, and Bei Yu. Context-based contrastive learning for scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2022.
- [208] Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 61:348–360, 2017.
- [209] Yabin Zhang, Hui Tang, and Kui Jia. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *Proceedings of the european conference on computer vision (ECCV)*, pages 233–248, 2018.
- [210] Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-to-sequence domain adaptation network for robust text image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2740–2749, 2019.
- [211] Jinyuan Zhao, Cunzhaoh Shi, Fuxi Jia, Yanna Wang, and Baihua Xiao. Document image binarization with cascaded generators of conditional generative adversarial networks. *Pattern Recognition*, 96:106968, 2019.
- [212] Zhao Zhong, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Handwritten chinese character recognition with spatial transformer and deep residual networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 3440–3445. IEEE, 2016.
- [213] Jing Zhou, Yanan Zheng, Jie Tang, Jian Li, and Zhilin Yang. Flipda: Effective and robust data augmentation for few-shot learning. *arXiv preprint arXiv:2108.06332*, 2021.
- [214] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [215] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.





