Segmentation, Recognition and Indexing of Cham characters in Cham documents

Tien Nam Nguyen

LA ROCHELLE UNIVERSITÉ

École doctorale Euclide

Laboratoire Informatique, Image, Interaction (L3i)

THÈSE présentée par :

Tien Nam NGUYEN

soutenance le : **12 Juillet 2023**

pour obtenir le grade de : **Docteur de La Rochelle Université**

Discipline : **Informatique et Applications**

Segmentation, Reconnaissance et Indexation de caractères dans les documents CHAM

CLOPPET FLORENCE	PROFESSEURE	Université Paris Cité	Rapporteur
TABBONE ANTOINE	PROFESSEUR	Université de Lorraine	Rapporteur
RAMEL JEAN-YVES	PROFESSEUR	Université de Tours	Examinateur
LEMAITRE AURÉLIE	MAITRESSE DE CONFÉRENCES	IRISA - Rennes	Examinatrice
SCHWEYER ANNE-VALÉRIE	CHARGÉE DE RECHERCHE	CNRS	Examinatrice
LE THI-LAN	PROFESSEUR ASSOCIÉ	Institut Polytechnique de Hanoi	Co-Directrice
BURIE JEAN-CHRISTOPHE	PROFESSEUR	La Rochelle Université	Co-Directeur


La Rochelle
Université

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AG Attention Gated. vi, 68, 77, 81

ALCM Adaptive Local Connectivity Map. 31, 32

BAM Bottleneck Attention Module. vi, 69, 70, 77, 81

BGRU Bidirectional Gate Recurrent Units. 137

BiLSTM Bidirectional Long-Short-Term-Memory. 42

BLUE Bilingual Evaluation Understudy. 135–138, 140

CER Character Error Rate. 134, 135, 137, 138, 140

CNNs Convolutional Neural Networks. v, 14, 15, 21, 39, 41, 43, 66, 72, 126, 128, 137, 141

CRNN Convolutional Recurrent Neural Network. v, 40

CTC Connectionist Temporal Classification. 40, 44

DCT Discrete Cosine Transform. 18

DIA Document Image Analysis. v, xxi, 6, 8–12, 35, 46–48, 57, 61

DR Detection Rate. 108

FCN Fully Convolutional Network. v, 15, 33, 34

FM F-Measure. 108, 109

GAF Global Attention Fusion. 71, 81

GAN Generative Adversarial Networks. 21

GDT Gray-level Distance Transform. 94

GM Gradient Map. viii, 94–96

GRU Gate Recurrent Units. 138, 141

HCR Handwritten Character Recognition. 35

ICA Independent Component Analysis. 18, 65

kNN K-nearest neighbors. 36, 38

LCG Labeling Cutting Grouping. viii, ix, 94–97, 118

LSTM Long-Short-Term-Memory. 25, 46

MLP Multi Layer Perception. 15, 20, 39, 43, 66

MSE Mean Square Error. 76

MST Minimum Spanning Tree. viii, xiii, 102, 103

NLP Nature Language Processing. 41

NPW Neighborhood Pixels Weights. 36–38

OCR Optical Character Recognition. v, 9, 10, 35, 42, 122, 146

PCA Principal Component Analysis. 18, 19

PSNR Peak Signal-to-Noise Ratio. 76–78

RA Recognition Accuracy. 108, 109

RNN Recurrent Neural Networks. 39–41, 43, 46, 126, 128, 141

SCPs Significant Contour Points. 24, 25

SOTA State-of-the-art. 124, 128, 136

SSIM Structural Similarity Index Measure. 76–79

SVM Support Vector Machine. 23, 36, 38

WER Word Error Rate. 134, 135, 137, 138, 140

Acknowledgements

I would like to use this page to express my deepest gratitude to many peoples who have supported and shared idea during my work.

First and foremost, I would like to thank my supervisors Prof. Jean-Christophe BURIE and Assoc. Prof. Le Thi Lan for their invaluable guidance, and insightful feedback, constructive comments throughout my research. Their expertise and dedication have helped shape the direction of my work and have been a constant source of inspiration.

I would like to thanks Prof. Florence CLOPPET and Prof. Antoine TABBONE to have accepted to review this dissertation and for sharing constructive comments and very insightful feedback. I am also grateful to Prof. Jean-Yves RAMEL and Aurélie LEMAITRE for evaluating my research work.

I would like to extend my appreciation to the participants of this project. Especially, Anne-Valerie SCHWEYER, coordinator of the ANR ChamDoc Project, for the unwavering support, creating the best condition and for accepting me in this project. Her supports have been crucial in the orientation and development my thesis.

My gratitude extends to all my friends and my colleagues: Van, Hai, Cong, Bao, Dipendra, Ibrahim, Benson, Muzzamil, Dom, Mélanie, Kathy at L3i, Zuheng, Julien, Jasmine, Imane,

Kacem at Open Space 121, Isabelle and Jenifer from Ecole Doctorale, Prof. Hai, Prof. Hai, Nam from MICA and the old colleagues from TSDV: Duc, Thanh, Tam. Their support, advises, and the shared experiences have made the process more enjoyable and meaningful.

Lastly, I am deeply indebted to my father Nguyen Tien Dong, my mother Nguyen Thi Thanh Hao and my sister Nguyen Thi Thu Hang for their unwavering love, encouragement, and support throughout my academic pursuit. Their belief in my abilities and sacrifices they made to facilitate my education are immeasurable. I am also to thankful to many people in my family: my brothers, my sisters, my aunts, my uncles. A lot of my strength come from them.

While it is not possible to mention everyone who has contributed in some way to this thesis, I am sincerely grateful to the many peoples, students, citizen that I had the chance to meet in La Rochelle: Miha, Hanh, Cong's family, My Anh, Thu, Nhung, Thomas, Ly, Ha, Long, Hang, and Hong Anh, Cindy, Estelle, Tu (from Bordeaux), many young students from Excelia: Van, Thu, Dung, for their encouragement and their help.

Abstract

The study of cultural heritage has attracted many researchers in recent decades. Historical handwritten documents are important evidence for understanding historical events and especially those of vanished civilizations

Since the demise of Champa kingdoms during the 19^{th} century, the *Cham* language that originated and developed from 2^{th} century, is no longer really used among the descendants of the Champa. The lack of transmission of knowledge and documents of the Cham culture makes the study of this language difficult for epigraphists and historians.

Therefore, the ANR CHAMDOC project aims to preserve and provide tools for studying the Cham language. In this thesis, we focused on the analysis of two types of Cham documents namely: inscriptions, which were engraved on stone steles, from 6^{th} to 15^{th} century and manuscripts dating from the 18^{th} century. Some work on the digitization of inscriptions has started but no study has really been carried out. The collection of manuscripts, for its part, has never been exploited. These two document collections offer many challenges for the scientific community. During this work, we propose a complete pipeline for the automatic processing of these documents. This is based on different Document Image Analysis (DIA) techniques. The challenges encountered come from the characteristics of the documents themselves, but also from the linguistic specificities of Cham. An analysis of these characteristics has been carried out in order

to propose solutions adapted to inscriptions and manuscripts.

More specifically, we focused our efforts on three main tasks: the denoising of the inscription images to improve their visual quality, the segmentation of the text lines, and finally the transliteration of the text lines :

- For the first task, we proposed a model based on an encoder-decoder combined with an attention mechanism to not only remove unwanted parts but also to improve the visual quality of the document.
- For the text line segmentation task, different strategies using deep learning, traditional image processing techniques or a combination of both have been proposed to adapt to the specificities of the documents.
- For the text line transliteration, a two-step approach has been implemented to recognize and then correctly transcribe Cham scripts into Latin characters.
- Moreover, in order to evaluate the proposed methods at each stage of the pipeline, several annotated datasets have been created with the help of our expert.

The experiments, carried out at each stage of the pipeline, show the relevance of the proposed methods.

Keywords: Champa Kingdoms, Cham Manuscripts, Cham Inscriptions, Document Image Analysis, Denoising Image, Visual Quality Improvement, Text Line Segmentation, Transliteration.

Résumé

L'étude du patrimoine culturel a attiré de nombreux chercheurs ces dernières décennies. Les documents manuscrits historiques sont des preuves importantes pour comprendre les événements historiques et en particulier ceux des civilisations disparues.

Depuis la disparition des royaumes Champa au cours du $19^{ème}$ siècle, la langue *Cham* qui est née et s'est développée à partir de $2^{ème}$ siècle, n'est plus vraiment utilisée chez les descendants des Champa. Le manque de transmission des connaissances et des documents de la culture Cham rend, l'étude de cette langue,.difficile pour les épigraphistes et les historiens.

Par conséquent, Le projet ANR CHAMDOC vise à préserver et proposer des outils pour étudier la langue Cham. Dans cette thèse, nous nous sommes concentrés sur l'analyse de deux type de documents Cham à savoir : des inscriptions, qui ont été gravées sur des stèles en pierre, du $6^{ème}$ au $15^{ème}$ siècle et des manuscrits datant du $18^{ème}$ siècle. Quelques travaux sur la numérisation des inscriptions ont été menées mais aucune étude n'a vraiment été réalisée. La collection de manuscrits, quant à elle, n'a jamais été exploitée. Ces deux collections de documents offrent de nombreux défis pour la communauté scientifique.

Au cours de ces travaux, nous proposons un pipeline complet pour le traitement automatique de ces documents. Celui-ci est basé sur différentes techniques d'analyse d'images de documents. Les défis rencontrés proviennent des caractéristiques des documents eux-mêmes, mais aussi des

spécificités linguistiques du Cham. Une analyse de ces caractéristiques a été menée afin de proposer des solutions adaptées aux inscriptions et aux manuscrits.
Plus précisément, nous avons concentrés nos efforts sur trois tâches principales : le débruitage des images d'inscriptions pour amélioration la qualité visuelle de l'image, la segmentation des lignes de texte, et enfin la translittération des lignes de texte:

- Pour la première tâche, nous avons proposé un modèle basé sur un encodeur-décodeur combiné à un mécanisme d'attention pour, non seulement, supprimer les parties indésirables mais aussi pour améliorer la qualité visuelle du document.
- Pour la tâche de segmentation des lignes de texte, différentes stratégies utilisant l'apprentissage en profondeur, des techniques traditionnelles de traitement d'images ou une combinaison des deux ont été proposées pour s'adapter aux spécificités des documents.
- Pour la translittération des lignes de texte, une approche en deux étapes a été mise en œuvre pour reconnaître puis transcrire correctement les scripts Cham en caractères latins.
- De plus, afin d'évaluer les méthodes proposées à chaque étape du pipeline, plusieurs jeux de données annotées ont été créés avec l'aide de notre expert.

Les expérimentations réalisées à chaque étape de la chaine de traitements montrent la pertinence des méthodes proposées.

Keywords : Royaumes Champa, Manuscrits Cham, Inscriptions Cham, analyse d'images de document, Débruitage Amélioration de la Qualité visuelle des images, Segmentation de ligne de texte, Translittération.

CHAPTER 1

Introduction

1.1 Context of the research work

We are now living in the 21^{st} century where people's lives are gradually developing and complementing each other in different aspects: economics, politics, and society. The current results may not come instantly but it continuously formed and developed over a long period. Our lives today have been influenced by several historical events which are recorded in many types of documents. Some of them contributed significantly while others did not so much. There is a lot of exciting information to explore especially for curious people and historians. For instance, it is interesting to see how an event in one country can influence the life, culture, and people of the other countries; how the communities/kingdoms are established then dissolved; and the evolution of policy, social class, and culture of each country. Such information has partial or complete answers, however, many things are still unknown and unanswered until now because of the missing information or links between the resources. Therefore, on the one hand, studying, analyzing, and exploring the events of the past in the process of further obtaining information. On the other

hand, it is also an interesting way to improve the present life by previous knowledge.

In parallel, language is one of a ways to communicate in any culture. Language provides the ability to understand thoughts, and ideas, and build relationships, so that people can understand each other. Starting from the basic characters, over time, they gradually expanded, and complemented each other to form different languages. There are about approximately 6500 languages in the world, and each of them has its own story. While the main languages such as English, Mandarin, Hindi, French, Spanish, and Arabic were well-studied and are still used, some languages such as the ones found in historical documents are little studied or even forgotten. Particularly, for some kingdoms that have disappeared like Champa whose language (Cham) is studied in this dissertation, the access to the information is quite difficult. In fact, no Vietnamese people can read old Cham inscription documents from 6^{th} to 15^{th} which are written in old Cham script or Sanskrit script and only a few people can read partially Middle Cham (from 16^{th} to 18^{th}). t

Considering these two above aspects mentioned (language and invaluable information), historical documents are the indispensable evidence to understand how the ancient societies functioned. Indeed, these historical documents can be considered cultural heritage because of their indispensable value not only from the culture but also of the society and the way of life of the time concerned. Although a large amount of documents is available, their study is quite limited by two things.

First, these documents are often scattered because the process to collect them is very time-consuming and costly. Moreover, the owners are not always aware of the importance of these documents. Some documents have been burdened or crumpled while other documents have been damaged due to the climatic conditions. Due to the lack of responsibility and awareness, their preservation has not received enough attention, leading to partial or even complete loss of documents.

Secondly, over time, the number of resources for some languages has decreased, which has reduced the number of people able to read and understand them (mainly historians, archaeologists, and a few citizens). The transfer of knowledge then becomes complicated, making it difficult to transmit it to future generations.

In this work, we aim to preserve the Cham handwritten historical documents collected in Vietnam. The corpus is constituted of inscriptions engraved on stone steles or monuments and manuscripts on sheets of paper or palm leaves. These documents are written according to an Indic system but adapted to two different languages: Sanskrit and Cham, which have been used from the very early centuries AD in Champa. Research on the Cham language is quite limited due to the difficulty encountered when accessing these documents. Programs for digitizing these documents capable of analyzing, classifying, transliterating/automatically translating these documents are rare, making their use and study almost impossible. The catalogs of epigraphic discoveries are available Cœdès (1908), Schweyer (1999), but the accessibility to the texts remains restricted (for example, Finot (1995), Griffiths et al. (2012)]). Moreover, diachronic linguistic studies on Cham languages are non-existent and studies on modern Cham languages remain very confidential Thurgood (1999), Brunelle (2008) to Brunelle (2001), mainly because of the lack of accessibility to written sources. To our knowledge, there is no research work available on the automatic analysis of Cham writings. For the inscriptions, one of the pioneer studies stems from the research work [1] carried out in collaboration with the École française d'Extrême-Orient (EFEO) and the Institute for the Study of the Ancient World (ISAW) at New York University from 2010 to 2012. The research work has only been conducted on the Cham inscriptions. Few results have been obtained from this study such as the construction of a dataset including :

- the place where the inscription had been found;
- the place where it was currently located (if it had been moved after being discovered);
- the language(s) used in it;
- its date;
- availability of reproductions of it in public libraries;
- bibliography of publications about the inscription.

For the manuscripts, through the Endangered Archives program, the British Library has funded projects to digitize Cham manuscripts on latanier leaves or on paper in order to save them.

[1] http://isaw.nyu.edu/publications/inscriptions/campa/

However, no one is able to use these documents (now available online) [2]. Overall, these research works are only done by focusing mainly on digitization, and few of them with the linguistic aspect. Even if the digitized images are available online, to our knowledge, most researchers are unable to decipher them and no program has attempted to extract and index the content to facilitate the work of historians, archaeologists, etc. Furthermore, these linguistic researches are crucial for conducting the research works but if they are interesting, they are quite limited. It is impossible to manually explore and analyze a large number of documents (about 250 inscriptions in Sanskrit and in Cham and millions of pages of manuscripts in Middle Cham) due to the human limits. Collaboration with researchers in Computer Sciences was therefore considered not only to develop approaches to accelerate document processing but also to take advantage of artificial intelligence to deeply understand and analyze the underlying language. This collaboration is a mutual interaction. On the one hand, computer scientists can build automatic tools for processing documents while on the other hand, linguistics can provide their valuable knowledge about language which helps computer scientists recognize the encountered problems, and then adjust them to gradually enhance the performance.

Following these observations, CHAMDOC project has been created. This project addresses two issues: the difficulty of accessing sources and the creation of automatic recognition and analysis programs for syllabic alpha writing. It fits into the continuity of work undertaken in 2018 by some members of the consortium as part of support for the international mobility (SMI) funded by the Centre National de la Recherche Scientifique (CNRS) on the feasibility study of a program for automatically reading old Champa inscriptions. Our goals revolve around the scientific axis “Culture, Creations, Heritage” of ANR Program. In addition to focus on the preservation of these documents, we also aim to develop a complete workflow to automatically process, analyze, recognize then index them and give access easily to their content. This is a multi-disciplinary project, with an international collaboration, taking advantage of the knowledge of experts in Cham language and Computer Science. This work is carried out by four research institutions. On the one hand CASE (Centre Asie du Sud-Est, Paris) and LSS (Laboratoire des Structures Sonores, Ottawa) mainly work on linguistic problems, and on the other hand, L3i (Laboratoire Informatique Image Interaction, France) and MICA (Multimedia Information Communication

[2] https://eap.bl.uk/project/EAP698/search

and Applications, VietNam) which work on the development of innovative approaches in computer science. This work is supported by the French National Research Agency (ANR) in the framework of the ChAMDOC Project, n°ANR-19-CE27-0018-02.

1.2 Objectives of the work

Cham is the language of the Champa people whose history has marked the Vietnam area since the 2^{th} century. Originally, most of the documents were written on the surface of the stone, called hereafter "inscription". On the inscriptions engraved between the 6^{th} and 17^{th}, there are only two languages (Sanskrit and Cham), but many different scripts have been used for writing over time. Today, only a handful of specialists in the world can read these inscriptions and when they disappear, no one will be able to read these historical documents for lack of transmission of knowledge. Similar to the inscriptions, the manuscripts which were written by Middle Cham, there are also a few people who can completely understand the meaning of these documents. This can be explained by the fact of the disappearance of Champa kingdoms. In fact, since 1832, Champa's kingdoms have been part territory of the Nguyen dynasty (South of Vietnam's area now). The language of the kingdom has still been used in the community after this colonization. However, due to the policy and the standardization of the language by the Nguyen dynasty whose writing is quite different from the Cham scripts (Nguyen dynasty utilized many kinds of scripts such as Chinese scripts, i.e. pictograms Han Nom, and Quoc Ngu romanized-scripts after 1920). It causes the use of Cham language to become limited and then gradually disappear.

The complexity of Cham language (Austronesian) and scripts (alpha-syllabic) makes their transmission almost impossible and the progressive assimilation of the Cham populations within modern Vietnam raises fears of the extinction of this culture. Indeed, the Cham themselves are no longer able to read the old and middle Cham documents. To prevent the disappearance of these alpha-syllabic writings, the ChAMDOC project aims to develop methods based on artificial intelligence techniques for processing, analyzing and recognizing, indexing these document images. Our main work is to enable automatic reading of the sources and their indexing to allow wider exploitation and further exploration of Cham culture. Specifically, these goals are

expressed mainly around the following lines:

- The first objective is related to the long-term preservation and enhancement of valuable Cham documents. This is the construction of standard processing of these documents which includes several tasks such as: storing, indexing, digitization, and ground truth creation.
- The development of a complete workflow for the automatic processing of these documents. The workflow based on the standard Document Image Analysis pipeline includes several tasks such as pre-processing, layout analysis, recognition, and transliteration by considering the characteristics of the Cham documents.
- The indexing of words to allow the creation of semantic links between documentary sources. Through indexing, users can easily access and instantly search for relevant instances in documents.

In this dissertation, we mainly focus on the second task: the design and development of innovative methods and tools for the extraction and analysis of characters from these ancient written documents. Our scope of work consists of two collections: the Cham inscriptions and the Cham manuscripts, making them accessible for both two types of documentary sources.

1.3 Contributions of the thesis

The contributions of the thesis are listed as follows :

1. A new model for image denoising based on an encoder-decoder with symmetrically skip connection and residual block was proposed. In this model, the global attention fusion was introduced to learn the relevant regions in the image. The proposed method can not only remove unwanted parts in the image but also enhance the visual quality of the Cham inscriptions.
2. An effective text line segmentation method based on the combination of different cost functions considering the characteristics of the Cham inscriptions was introduced. The proposed method was evaluated on our own dataset of Cham inscriptions and Cham

manuscripts as well as the dataset of Medieval Manuscripts Simistira et al. (2017). Experimental results confirm the robustness of the proposed method compared with different deep learning approaches.

3. A new scenario for text line transliteration by using a two-step approach where similar characters are first considered as unique characters, then we use a transformer model which considers both visual and context information to adjust the prediction when it concerns similar characters to be able to distinguish them. Extensive experiments have been conducted to compare and evaluate our approach and some methods of the literature on the Cham manuscripts dataset.

4. Creation of three image datasets named **Denoising-Inscription-Cham**, **Textline-Inscription-Cham**, **Transliteration-Manuscript-Cham**. These datasets are constructed from two types of Cham documents (the Cham inscriptions and the Cham manuscripts). The documents have been collected and annotated for evaluating the different tasks developed for the analysis of Cham documents.

    - **Denoising-Inscription-Cham**: the dataset consists of 190 text line images of inscription prepared for image enhancement method Nguyen et al. (2021b).
    - **Textline-Inscription-Cham**: a set of 395 text line images from 26 inscriptions images for the text line segmentation Nguyen et al. (2022). This dataset is publicly available [3]. Moreover, a dataset of **Textline-Manuscript-Cham** from 600 manuscript images is also used for qualitative evaluation.
    - **Transliteration-Manuscript-Cham**: a set of 4507 text line images of manuscripts built for evaluating text line transliteration method.

1.4 Organization of the manuscript

The rest of this dissertation is organized as follows.

In Chapter 2, we introduce the Champa communities, its languages (Cham), and two collections

[3] http://l3i-share.univ-lr.fr/2022CHAMDoc/CHAMDoc.html

of documents: the Cham inscriptions engraved on stone steles and the Cham manuscripts. For each collection, the history and the range of documents as well as the features of Cham language are also presented. The disadvantages and the advantages when applying DIA on each dataset are also analyzed.

In chapter 3, we first present an overview of DIA which its standard steps. For each step in the general workflow, the well-known approaches and their evolution are presented. Their weakness, strengths, and applications are also analyzed. After that, we present the pipeline applied to Cham documents.

In Chapter 4, we define the problem encountered when pre-processing the inscriptions. Then, the details of the proposed method for image enhancement are presented. Extensive experiments based on an ablation study and a comparison with other approaches of the literature on our Cham inscription dataset are carried out. Moreover, the initial weakly supervised training has been tested and evaluated.

In Chapter 5, we present our contribution to the text line segmentation problem on the Cham inscriptions dataset and an adaptation for the Cham manuscripts dataset. Furthermore, an application of the Medieval Manuscripts is presented. The experiments as well as the comparisons with different methods are also detailed.

In Chapter 6, we detail the text line transliteration by considering this task as a recognition problem. After that, we describe the proposed approach for text line transliteration, an ablation study with different modules, and a comparison with different methods of the literature.

Finally, the summary of the work, conclusions, and future works are detailed in Chapter 7.

CHAPTER 2

State-of-the-arts

2.1 Introduction

In this chapter, we present the strategies found in the literature for each stage of the Document Image Analysis (DIA). DIA refers to algorithms and techniques that are applied to images of documents to obtain a computer-readable description from pixel data Kasturi et al. (2002). Based on this definition, DIA, can be broken down into two main components of DIA consisting of document analysis and document recognition.

- Document analysis is the process of identifying and segmenting the important components of a document. This involves splitting the document into sub-components such as text, images, and tables, and analyzing their layout and structure. The objective of this process is to extract meaningful information from the document that can be used for further analysis or indexing.
- Document recognition is the process of recognizing the components detected during the analysis phase. For example, it can involve the use of Optical Character Recognition

(OCR) which converts scanned images of text into machine-readable formats. The objective of this process is to convert the document into a format that can be easily used for other tasks such as searching, indexing, and retrieval.

To achieve better results for the two components mentioned above, additional pre-processing and post-processing steps are often used. For example, some documents may require additional pre-processing steps to correct image distortion Amin and Fischer (2000) or enhance the contrast between text and background Likforman-Sulem et al. (2011), Nomura et al. (2009). Other documents may require an additional step for the recognition algorithms Kesiman (2018) to handle non-standard fonts or languages. In this work, we provide a study of related works for each component in DIA, by analyzing their strengths as well as their weaknesses. After that, a review of software available for DIA that can be used directly for the new corpus is also presented.

2.2 Overview of document image analysis pipeline

Document Image Analysis (DIA) is a complex task due to the number of elements that may appear in the document such as paragraphs, decorations, texts, images, tables, etc. Over time, several stages within the DIA have emerged, and the original task has been split into sub-tasks in which the output of the previous step is the input of the next step. Figure 2.1 shows an overview of the entire document image analysis pipeline and highlights some major tasks for each component.

The first step is often related to the *digitization* task. Document digitization is the process of transforming physical documents into digital format for easier storage, access, and retrieval. While printed documents are already in computer-readable digital format, most historical documents must use the process of digitization to convert them into readable format.

After the digitization process, different *pre-processing* tasks can be applied such as: image binarization, image denoising, and image enhancement. These steps are generally applied with the following objectives: enhancing the quality of documents, removing unwanted parts, and converting the image to binary format.

The next step concerns the *analysis* stage. Two main tasks can be considered: First, *logical*

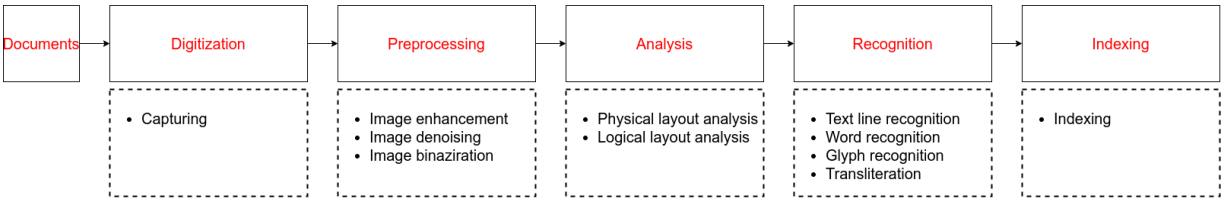


Figure 2.1: Overview of the entire pipeline in Document Image Analysis (DIA)

layout analysis aims to identify/assign the regions of the pages (in case of multiple pages) or different structures in the case of multi-media documents (containing both images and texts). Then the *physical layout analysis* task involves segmenting the different components existing in the documents such as text-block (paragraph), text-line, word, glyph, and character.

The *Recognition* step aims to recognize the segmented components. In historical document analysis, this stage includes text line recognition, word recognition, and glyph/character recognition. Transliteration can also be considered as a recognition task in which the system not only recognizes the text but also needs to transcribe it to another writing system. Finally, it is the *indexing* step. It is related to organizing and storing the metadata extracted from documents to quickly retrieve relevant information.

As the *indexing* step is beyond the scope of this thesis, in the following sections, we address quickly the problem of digitization and then we mainly focus on analyzing different approaches proposed for pre-processing, analysis, and recognition steps.

2.3 Digitization

The digitization task is an important step and must be applied carefully and precisely to keep the quality of the input documents, otherwise, some distortions may appear and/or some information may be lost, which complicates the following processing steps. Typically, the digitization task, considering the historical documents, can be done using a standard digital camera, as in the works on Balinese palm leaf manuscripts Kesiman (2018), Medieval manuscripts Simistira et al. (2017), Khmer palm leaf manuscripts Valy (2020), Stain Gall Database Fischer et al. (2011), IMPACT dataset Antonacopoulos et al. (2013) or IHR-NomDB Vu et al. (2021). However, with some specific datasets like our collection of Cham inscriptions where letters are engraved on

stone surfaces, the use of a standard digital camera cannot capture the correct resolution of the documents. Thus, a rubbing process is used. It consists of first sticking paper on the surface of an object (stone, bricks, or metal) and rubbing it. Then this paper will be scanned through a scanner machine to export the document in a digital image format. However, this approach raises some problems. Two main issues need to be considered: the physical input documents may be damaged and the quality of the output documents can not be guaranteed. Distortions, extra unwanted parts, or missing parts of the input image may appear if the rubbing process is not done carefully. This can lead to a reduction in the performance of the DIA. Moreover, this process can alter the stones due to the friction with the paper. Therefore, controlling the process must be considered.

2.4 Pre-Processing techniques

The preprocessing step is proposed to prepare data and so provide better input for further steps. The results of this step have a huge impact on the whole pipeline because errors can propagate to the next tasks. It is, therefore, an important component of the DIA pipeline. In general, image binarization is often used at this step to transform the original image into a binary image with only black and white pixels. However, for some types of documents such as historical documents, where quality of documents is often degraded, using standard binarization on the image may not work well. Thus other methods to improve the quality of documents need to be used such as image enhancement or image denoising.

2.4.1 Image binarization

Image binarization or image thresholding is the kind of approach to transforming a grayscale input image into a binary image with pixel values changing from 256 to 2 (0 for black pixels, while 1 represents white pixels). The binarization task can be addressed according to three types of approaches: global thresholding, local thresholding, and deep learning methods. The comparison of several methods of binarization applied to old documents is presented in Khurshid et al. (2009), He et al. (2005).

Global thresholding

The approach proposed by Otsu in Otsu (1979) (called Otsu's Thresholding) is one of the pioneering works in this category. Pixels whose value is greater than a given threshold will be assigned the value 1 otherwise they will be assigned the value 0. This threshold can be set manually but it will not be suitable for other images. Therefore, statistical pixel analysis is used to determine this threshold. The underlying principle of this selection is based on the values of the histogram of the image. This method assumes that the histogram is bimodal and presents two distinct Gaussian distributions in the pixel value distribution. The chosen threshold is better when its value makes it possible to split this histogram into two parts. Figure 2.2 illustrates the effect of changing the threshold on the binarization results. The extensions of the original Otsu's method are introduced in the works of Liao et al. (2001) and Barron (2020). In Barron (2020), additional statistical measurements are used to select the threshold. Instead of using the previous assumption (bimodal histogram) about the histogram, the threshold is chosen by maximizing the joint probability of all pixels in which each pixel is generated from a mixture of two probability distributions. The proposed approach outperformed various traditional image binarization methods as well as deep learning methods.

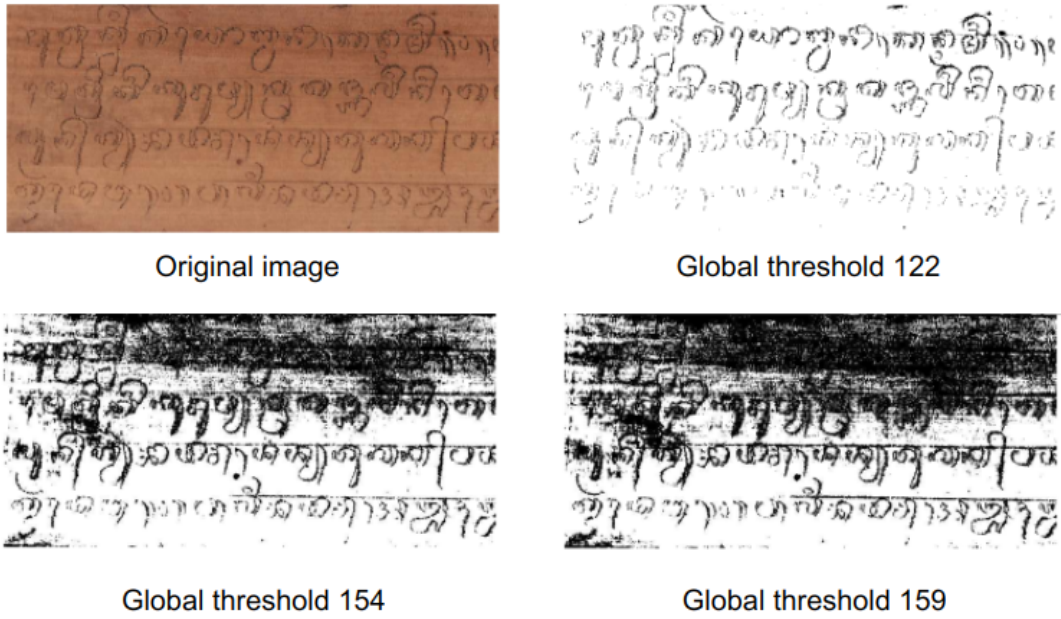


Figure 2.2: Comparison of binarization results with different global threshold values on Balinese Palm Leaf Manuscripts Kesiman (2018)

Local thresholding

Instead of using a single threshold on the whole document, which can produce unsatisfactory results when the input images do not verify the assumption of a bimodal histogram, a thresholding technique local is proposed to work on local areas. Niblack's Niblack (1985), Sauvola's Sauvola and Pietikäinen (2000), Wolf's Wolf et al. (2002), Feng's Feng and Tan (2004), NICK's Khurshid et al. (2009) are the outstanding works on this category. In the work of Niblack Niblack (1985), the threshold value is computed as the sum of the mean and standard deviation of the neighborhood in the sliding window. The disadvantage of this algorithm appears when it is applied to non-textual areas, it generates a lot of noise, especially when the backgrounds have discontinuities. An extension of Niblack's is proposed in the work of Sauvola Sauvola and Pietikäinen (2000). The standard deviation is re-calculated according to two user-defined parameters R and k (related to the variations in local intensity). By using this strategy, the problems of non-uniform illumination and changing background are solved, but it is still a problem when text and non-text areas have close pixel values. To address the issues of Sauvola's algorithm, a normalization of the contrast and the mean grey value of the image is proposed in Wolf et al. (2002). This strategy provides a huge improvement in comparison with the two previous methods but the performance is degraded when the image contains noisy patches (sharp change) leading to an incorrect calculation of the parameters. In general, the local adaptive thresholding is better than the global threshold but the main issues with this approach are the determination of the parameters such as window stride, pre-defined parameters, or the structure of the input image. This method will be difficult to adapt to our collection due to the great variability of the documents in terms of intensity, and contrast.

Deep learning network

Deep learning approach is another strategy used to handle the huge variations in big datasets. The first work using deep learning for this task is proposed in Pastor-Pellicer et al. (2015). By considering the binarization problem as a classification task at the pixel level, the authors proposed a CNNs by carefully setting parameters such as window input size (9x9), kernel size (6x6), first sampling layer (2x2), second sampling layer (4x4), the topology of the final hid-

den layers (by two hidden layers of sizes 32 and 16). (See Figure 2.3). The proposed method

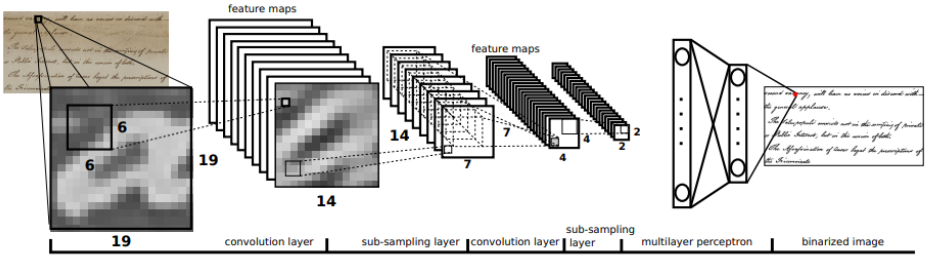


Figure 2.3: CNNs for document image binarization Pastor-Pellicer et al. (2015)

showed promising results in comparison with thresholding methods and Multi Layer Perception (MLP). To further improve the quality of output documents, multi-scale FCN Peng et al. (2017); Vo et al. (2018b) is proposed by combining information from different levels of features. In Vo et al. (2018b), the authors have observed that some details of the input image are lost at higher level features, while lower level features are useful in preserving the details of the foreground. Therefore, the integration of features extracted at different levels may lead to better performance. In detail, three deeply-supervised nets DSN_C3, DSN_4, DSN_C5 which contain respectively three, four, and five convolutional-layer groups are trained independently (using the same training data) (See Fig. 2.4). The final prediction is then determined from the prediction

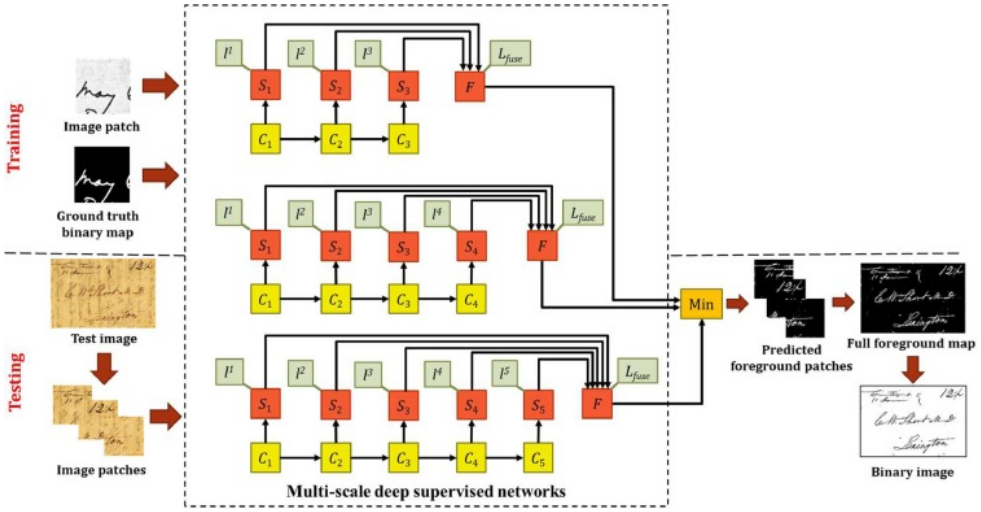


Figure 2.4: Hierarchical deep supervised network for document binarization proposed by Vo et al. (2018b)

of each network as shown in the following equation :

$$Y^p = min(Y_{fuse}^{DSN\_C3}, Y_{fuse}^{DSN\_C4}, Y_{fuse}^{DSN\_C5}) \tag{2.1}$$

where $Y_{fuse}^{DSN\_C3}$, $Y_{fuse}^{DSN\_C4}$, $Y_{fuse}^{DSN\_C5}$ are the predictions of each image patch obtained respectively from DSN_C3, DSN_C4, DSN_C5. Inspired by generative adversarial networks Goodfellow et al. (2020), in Zhao et al. (2019), the authors split the binarization problem into two stages. In the first stage, a cascade generator is used to generate binary images at different scales. In the second stage, another generator leverages different scale binarization images to produce the final results. To avoid using a pair of training data that are costly to set up, in the work of Tensmeyer and Martinez (2017), they apply CycleGAN Zhu et al. (2017) strategy which used unpaired data (no need to prepare the paired input and corresponding ground truth) to make the learned model more robust. Lately, a strategy using an iterative training model He and Schomaker (2019) is proposed to gradually improve binarization results.

2.4.2 Image denoising

The input document images may have many unwanted parts due to the storing condition that has damaged the documents and/or the digitization process that can reduce the contrast. In such cases, directly applying binarization is not effective since the process can introduce "noise" in the binary images. This issue can be considered as the image denoising problem which has been well-studied in computer vision. Usually, the traditional methods applied to image denoising problems use the characteristics of the noise present in the image.

Filtering based approaches

In the early, the approaches were based on image processing or signal processing methods. These methods can be organized into several categories such as spatial filters Pitas and Venetsanopoulos (2013); Tomasi and Manduchi (1998); Wiener et al. (1949) or transform domain Choi and Baraniuk (1998); Portilla et al. (2003); Ram et al. (2011). The denoising in the spatial domain is based on the observation that noise appears at high frequencies and low pass filters are adapted to eliminate or reduce this noise. These filters can be split into two categories: linear

or non-linear filters. The mean filter is the best-known linear filter. It replaces the value of each pixel with the average of all pixel values in a local neighborhood (usually an N by N window, where N = 5, 7, 9, etc.). The simplest non-linear filter is the Median filter which replaces the value of a pixel based on the median value of the pixels in a given neighborhood. A variant of the Median filter is the Weighted Median filter (Gaussian filtering). It consists of multiplying each pixel in the window by a corresponding weight, then performing a statistical sort, a new median value is assigned to the pixel. In general, the main limitation of these filter-based methods is that they are not robust to all types of noise. Each approach is adapted to a type of noise. Moreover, depending on the characteristics of the noise, effective filtering requires determining an adapted filter kernel size as well as an estimate of sigma when using a Gaussian kernel.

Other approaches are based on the pioneer work of Non-Local Means Buades et al. (2005). Specifically, the denoised image is estimated by weighting with similar patches on different locations of the image. These weights are calculated taking into account the similarity between the current patch and the neighboring patch and the distance between them. The computation steps are detailed below. Let's assume the input I, at a given pixel p, the weights with pixel q are computed as follows :

$$w(p,q) = e^{\frac{max(d^2-2\sigma^2,0)}{h^2}} \tag{2.2}$$

where σ is the standard deviation of the noise and h is the pre-defined parameter. d is computed as follows:

$$d^2(B(p,f),B(q,f)) = \frac{1}{(2f+1)^2} \sum_{j \in B(0,f)} (I(p+j) - I(q+j))^2 \tag{2.3}$$

where B(p,f) denotes a neighborhood centered at p of size (2f+1) x (2f+1). Then, the denoised image $\hat{I}$ is computed as follows:

$$\hat{I} = \frac{1}{C(p)} \sum_{q \in B(p,f)} I(q) w(p,q) \tag{2.4}$$

with:

$$C(p) = \sum_{q \in B(p,f)} w(p,q) \tag{2.5}$$

To handle the smoothness of the results, some suitable regularization methods can be used to enhance the quality of the denoised image such as L2 norm Tikhonov (1963), total variation regularization Rudin et al. (1992) in which they add constraints related to the gradient around the current pixel. Total variation regularization is defined as follows:

$$R_{TV}(x) = |\nabla(x)|_1 \tag{2.6}$$

where $\nabla(x)$ is the gradient of x.

As for the filter-based methods described above, the difficulty comes from the choice of hyper-parameters for regularization and similar patches which strongly influence the denoising performance.

In the transform-based approaches Choi and Baraniuk (1998); Portilla et al. (2003); Ram et al. (2011), the image is transformed into a new domain through Fourier transform, Wavelet transform or Discrete Cosine Transform where the characteristics of the noise and the clean part are different. After that, the clean image is obtained by performing the inverse transform after removing the coefficients whose values are supposed to represent the noises existing in the image. The main drawback of transform-based approaches is the determination of the right type of transform or wavelet bases to represent the data. A bad choice can lead to a decrease in the robustness of the method according to the data structure in the image.

Data driven approach

Data-driven methods are conventional approaches that are based on the statistical representation of the data. These methods try to represent a set of images into sub-components with some prior assumption and then remove redundancies in this representation which is estimated to be the noise present in the image. Methods based on Independent Component Analysis (ICA) Hyvärinen et al. (1998), Principal Component Analysis (PCA) Deledalle et al. (2011) and K-Single Value Decomposition (K-SVD) Lebrun and Leclaire (2012) are approached widely represented in this category. In Hyvärinen et al. (1998), the Independent Component Analysis is used. The authors assume that high-dimensional data can be represented as low-dimensional data that are statistically independent of each other. These independent components can be grouped together

to recover the clean image. Let's assume the noisy image y can be represented by the sum of a clean image x and a Gaussian white noise v.

$$y = x + v \tag{2.7}$$

In the first step, a set of noise-free images x are used to estimate a dictionary W

$$s = W.x \tag{2.8}$$

so that each component s_i of s is as sparse as possible. The densities of the s_i need to be modeled with a parameterization that is rich enough. The authors propose two ways to compute the density model $p(s)$

$$p(s) = Cexp(-as^2 - b|s|) \tag{2.9}$$

where a, $b > 0$ are parameters to be estimated, and C is a scaling constant.

or

$$p(s) = \frac{1}{2d}\frac{(\alpha+2)[\alpha(\alpha+1)/2]^{\alpha/2+1}}{[\sqrt{\alpha(\alpha+1)/2} + |s/d|]^{\alpha+3}} \tag{2.10}$$

with the estimation d and α defined as:

$$d = \sqrt{E(s^2)}; \alpha = \frac{(2-k) + \sqrt{k(k+4)}}{2k-1}$$

where $k = d^2 p_s(0)^2$. After that, the noisy patch $\hat{x}$ is then represented through the trained dictionary W with its coefficients $\hat{s}$. Shrinkage with different mechanisms such as hard thresholding, soft thresholding, non-negative garrote thresholding (non-linearity from estimation density [2.9, 2.10]) is applied to these coefficients. The inverse transform is performed on the truncate coefficients to get the denoised patches $\tilde{x}$.

$$\tilde{x} = W^T \hat{s} \tag{2.11}$$

Instead of using noise-free images like in Hyvärinen et al. (1998), in Deledalle et al. (2011), the authors used noisy patch images to learn an orthogonal basis matrix by performing a PCA. After finishing this step, the new patches are decomposed in an orthogonal basis matrix, where

the small coefficients are assigned to zeros.

The BM3D algorithm Dabov et al. (2007) is an impressive work that leverages the advantage of the spatial domain and the transform domain (See in 2.5). Similar parts are stacked into 3D blocks and then transformed into the wavelet domain. Simple thresholding or filter is then applied to remove noisy parts. The proposed method achieves very good performance compared to traditional and deep learning approaches. The problem with these traditional approaches is the choice of hyper-parameters and the estimation of the level of noise present in the image, which influences their robustness to different types of noise.

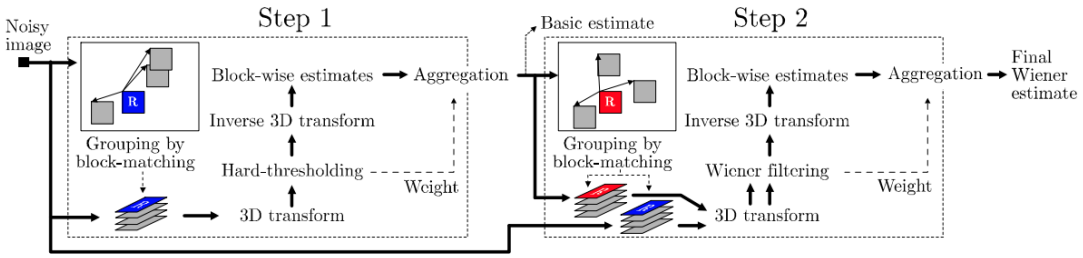


Figure 2.5: Proposed steps in BM3D Dabov et al. (2007)

Deep neural network

Although the traditional methods have demonstrated good results, especially BM3D which obtains very promising results in comparison with deep learning-based approaches, the main problem is that all the methods work under the assumption that noise is an additive white Gaussian noise with a given standard deviation. However, in real problems, this requirement is not always verified, so the noise needs to be represented by a more complex function. Most research then turned to deep learning-based approaches. This kind of problem can be considered an image translation problem (image to image) in which the model tries to transform the input from one domain to another domain through an association mechanism (noise/noise-free image) during the training process. One of the first works using deep learning approach is proposed in Burger et al. (2012). They use a Multi Layer Perception (MLP) to learn the mapping directly from a noisy image to a clean image. Different settings related to the number of hidden layers and the size of the layers have been conducted to evaluate the method. The two observations obtained from the training process are that larger patches and more complex networks performed better.

Based on this idea, many works have shown competitive results with very deep learning-based approaches that rely on the Convolutional Neural Networks (CNNs) such as: Mao et al. (2016), Zhang et al. (2017), Zhang et al. (2018). In the work of Mao & al Mao et al. (2016), the authors used a deep convolutional network composed of two parts: an encoder to learn the clean parts of the image and a decoder to reconstruct the original shape of the image. The skip connections are proposed to provide more information when reconstructing the image (see figure 2.6).

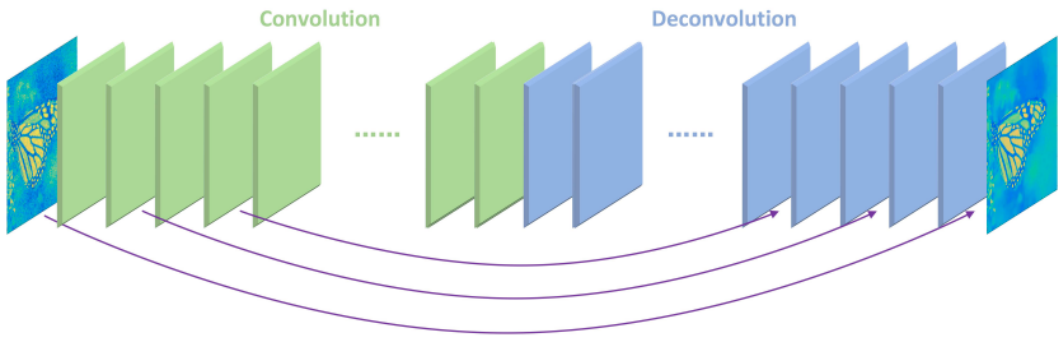


Figure 2.6: Encoder-Decoder with skip connections Mao et al. (2016)

Instead of directly mapping a noisy image to a noisy-free image, in Zhang et al. (2017) Zhang & al. proposed a model for learning the noisy space in the image, the noisy-free image can be obtained by subtracting the noisy image and the noise extracted from the model. Two other interesting approaches are the works proposed in Krull et al. (2018) and Lehtinen et al. (2018), where the training process is done without preparation of the ground truth (noisy-free image). In Lehtinen et al. (2018), the authors create two samples from the input image by adding two different noises. Parameters are tuned by comparing the reconstruction from the two samples above. In Krull et al. (2018), the authors propose a blind-spot network that forces the model to predict the hidden pixel in the center using the value of neighborhood pixels. The results do not outperform the standard supervision-based scenario approaches, but these methods have great potential for the dataset that does not have too many clean images. Inspired by the idea of Generative Adversarial Networks (GAN) Goodfellow et al. (2020), in the works proposed in Dumpala et al. (2019) and Souibgui and Kessentini (2020) the authors apply successfully generative modeling, in the context of historical handwritten document analysis.

2.5 Document layout analysis

Document analysis involves the process of analyzing and determining the structure of the documents O'Gorman (1993). The structure of a document is related to the sub-components present in the document. These components can be high-level elements such as titles, headings, sections, and captions and low-level elements such as text blocks, images, tables, lines, and words. Two types of analysis should be distinguished: logical layout analysis and physical layout analysis Doermann, Tombre, et al. (2014), O'Gorman and Kasturi (1995). Logical layout analysis consists in identifying/assigning labels to the document components while physical layout analysis consists in performing physical segmentation of document components.

2.5.1 Logical layout analysis

Logical layout analysis (document understanding) can be roughly classified into five main categories Dengel et al. (2014): rule-based methods, perception-based methods, learning-based methods, syntactic methods, knowledge-based methods.

- Rule-based methods Klink and Kieninger (2001), Lin and Xiong (2006): this approach is only used for particular domains due to prior assumptions about the documents. It is related to the use of a set of rules based on document-specific logical information such as font size, style, and relative position between the blocks to determine logical labels.

- Perception-based methods Eglin and Bres (2004), Lemaitre et al. (2008): is the type of approach based on how humans perceive visual information. People can classify these components through visual properties such as color, texture, and shape. This perception is facilitated by different mechanisms such as the functional description from texture primitives Eglin and Bres (2004) or the combination of features at the digital level (connected components and lines) and at the symbolic level (knowledge linked to elements extracted at the digital level) or at multi-resolution levels as in Lemaitre et al. (2008).

- Syntactic methods Aiello et al. (2002), Lin and Xiong (2006), Krishnamoorthy et al. (1993), Story et al. (1992): it is related to the similarity structure of the document and the syntax of the language. In Krishnamoorthy et al. (1993), syntactic analysis is used to

validate that the decomposition extracted from a block is valid with the assigned grammar. If it is valid, a label is assigned to this component otherwise an alternative grammar is used.

- Knowledge-based methods Aiello et al. (2002), Nagy et al. (2000): it is an approach based on the use of a knowledge base of domain-specific information. Knowledge can be used in different ways: document-generic knowledge (components of a document) and document-specific knowledge (text and font size format of document) in Aiello et al. (2002) or generic-knowledge, class-specific knowledge, and publication-specific knowledge as in Nagy et al. (2000). Once these knowledge bases are determined, a classifier is used to assign the labels.

- Learning-based methods (example-based training methods) Dengel et al. (1992), Staelin et al. (2007): in these approaches, no knowledge or pre-defined rules are prepared. All the rules will be learned through the corpus of documents used for the training. In Staelin et al. (2007), the system works in two steps: training and recognition. Each type of training document is used to feed the training system (using Support Vector Machine (SVM)) so that the system can link the words to the different metadata. Then, the cascade neural networks are used to classify the logical labels based on the extracted features and their probabilities determined from the SVM (See Fig. 2.7).

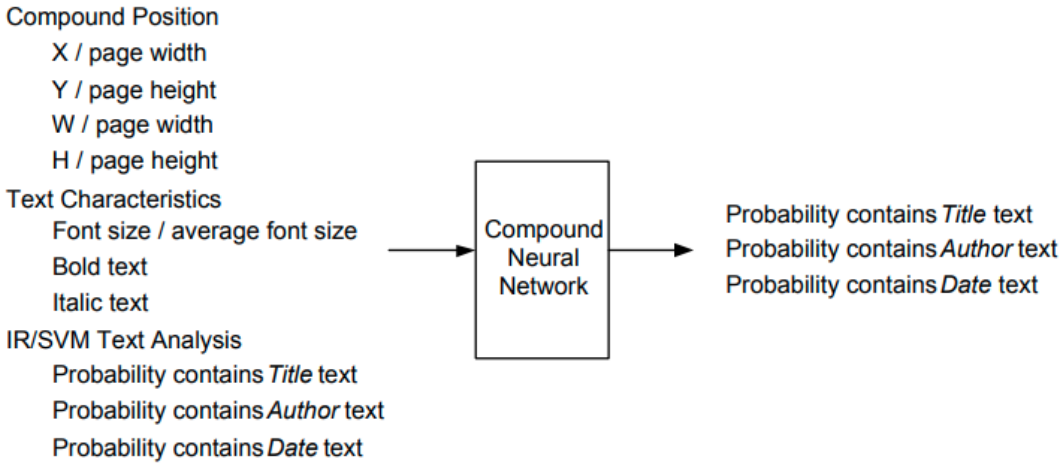


Figure 2.7: Inputs and outputs during training with the neural networks proposed in Staelin et al. (2007)

2.5.2 Physical layout analysis

Physical layout analysis can be done at different levels which mainly depend on the input of the recognition system. The most segmented components encountered in these approaches are isolated characters/glyphs, words, text-lines, text-block (paragraph). Here we summarize the two main tasks found in the literature: character segmentation and text line segmentation.

Characters segmentation

Characters segmentation is the lower level of page segmentation Kise (2014). This task consists of finding the isolated characters (glyph in some language Nguyen et al. (2019b)) from the sequence of characters. A representative survey about character segmentation can be read in Casey and Lecolinet (1996). The approaches for this problem can be classified into two categories:

- Heuristic-rule-based approaches: This kind of method uses character information to segment them such as font size, vertical space, or other related features. The pioneering work proposed in Hennis (1968) analyzes the characteristics of the characters based on the analog signal. If two consecutive columns of white bits are found, the decision is made on that basis. If no white columns are found within a distance determined by the measurements on the analog signal, segmentation is forced.

  Another subgroup of these methods involves the extraction and analysis of character features such as connected components, edge detection, texture analysis, and contour. In Hoffman and McCullough (1971), the authors extend the work presented in Hennis (1968) and propose a quasi-topological segmentation, which combines the extraction of different features: black-bit histogram (counting the number of black pixels in each vertical scan through a character), a horizontal stroke pattern (the number of times a given vertical scan encounters a part of the character) and a character measurement (measurement of upper and lower contour). While separating spaced characters is quite simple, the main problem comes when characters touch each other. In Strathy et al. (1993), the separating path is assumed to pass through a pair of Significant Contour Points (SCPs). In detail, a set of points corresponding to high curvature contours is selected from the contour points. After that, in the regions created by these points, 9 measures are applied (and ordered) to sort

the list of all possible matches. Finally, the separation path is generated by going from a starting SCPs to an ending SCPs.

Projection profile-based approaches are also used for this problem Kesiman et al. (2016a), Lu (1993). In Lu (1993), the character row is extracted by computing the horizontal projection profile. Then, the vertical projection profile is combined with the previous row of characters to segment the individual characters. The touching characters can be adjusted through the character recognition phase. In general, the drawback of this approach is the robustness when dealing with different styles of documents.

- Neural network-based approaches: While heuristic-rules-based methods work well for printed characters, separation performance is quite limited for handwritten characters, especially touching characters or cursive scripts. The use of neural networks is an alternative approach to tackle these cases Bae et al. (1998), Su et al. (2016), Zou et al. (2011). In Su et al. (2016), the authors consider the segmentation as a sequential classification problem, each region is labeled as character or non-character. More precisely, in the extraction phase, the characteristics of each image patch are extracted according to the relations between the pairs of adjacent features. Then, these features are fed into an Long-Short-Term-Memory (LSTM) to classify them.

Text-line segmentation

A text-line is a group of characters, symbols, and words that are adjacent, "relatively close" to each other and through which a straight line can be drawn (usually with horizontal or vertical orientation) O'Gorman (1993). Text-line segmentation is a crucial step in the whole pipeline of document image analysis especially for historical documents. The main purpose of text-line segmentation is to split the whole document image, line by line, to obtain a set of text-line images that will be used to feed the text recognition process. Some recent recognition methods can work at paragraph level Singh and Karayev (2021a), Wigington et al. (2018), so there is no need to split the input image into text line images. However, their applications are still very limited and depend on the layout of documents. Therefore, text line extraction is still widely used in the document analysis pipeline. Text-line segmentation methods can be mainly divided

into three main categories: top-down, bottom-up, and deep learning-based. A comprehensive survey for segmentation can be found in Eskenazi et al. (2017), especially Likforman-Sulem et al. (2007) tackles text line segmentation for historical documents.

- Top-down approach: In this approach, the pioneer works are based on the projection profiles method Manmatha and Srimal (1999). The projection profile consists of adding the number of pixels projected horizontally or vertically (depending on the studied direction). Vertical projection profile PP_I(y) and Horizontal projection profile (PP_I(x) are computed as :

$$PP_I(y) = \sum_{i=1:h} I(i,y) \quad ; \quad PP_I(x) = \sum_{i=1:w} I(x,i)$$

on image $I \in R^{h \times w}$. The peaks of the profile correspond to the text parts (lines of text) while the valleys correspond to the space between the lines (and vice versa if the text is black and the background is white). The basic projection profile usually contains local maximum points due to the noise or document structure that leads to wrong results. To avoid this problem, the projection profile is smoothed by a Gaussian filter Manmatha and Srimal (2002) or a moving average filter Valy et al. (2017). These methods work efficiently with documents that have a simple structure such as printed documents or documents with slightly skewed lines and/or curved lines. Different variations of this method have been proposed to deal with documents having skewed and curved lines Arvanitopoulos and Süsstrunk (2014), Boulid et al. (2015), Chamchong and Fung (2012) Valy et al. (2016). In Chamchong and Fung (2012), the authors propose an adaptive partial projection in which the documents are divided vertically into sub-columns, and the projection profile is then computed on the sub-columns. The points with the maximum value in each projection profile are considered as the candidate lines of each sub-column. Then, the candidate lines are matched through pre-defined rules Valy et al. (2016) or the distance between candidate lines Arvanitopoulos and Süsstrunk (2014). The problem that arises with this method is dealing with documents that contain many lines with different lengths or distances between the lines. The projection profile values do not allow to identification of candidate lines correctly.

Another approach belonging to the top-down category is the seam carving-based approach Asi et al. (2011), Arvanitopoulos and Süsstrunk (2014), Seuret et al. (2017), Alberti et al. (2019). Seam carving is first proposed by Avidan and Shamir (2007) for the image resizing problem. Specifically, a seam is a sequence of pixels that spans the entire height or width of an image and has the lowest cumulative energy value. So, we can resize the image by deleting these seams without distortion in the image. Adapting this idea to the text line segmentation problem, line separation (seam) is generated by minimizing the accumulated energy map computed, from the image, in which high energy values represent the text areas while low energy values represent background areas. In Asi et al. (2011), they propose a two-step seam carving to extract text lines. First, the seam seeds which mainly represent the base text line are generated from the seam-map generation in two passes. In the second step, they extract the separating lines by finding the lower and upper boundaries for each seam seed. In Arvanitopoulos and Süsstrunk (2014), a similar two-step strategy has been proposed. In the first step, the medial seam is computed from the partial projection profile. Let's the input $I \in R^{h \times w}$ and its edge image $S \in R^{h \times w}$. The whole image is split into r slices of width $m = w/r$. Then, horizontal projection profiles of each slice are computed as :

$$P_i^c = \sum_{j=k}^{k+m-1} S(i,j) \tag{2.12}$$

where $k \in (1, 1+m, 1+(r-1)m), c = 1, ..., r$

$$P^c = P_i^c, i = 1 : n \tag{2.13}$$

then the smoothed projection profiles is computed as follows:

$$P_g^c = g(P^c) \tag{2.14}$$

where g is the cubic spline smoothing filter. The merging of local maxima between slices

is done if the following matching conditions are verified:

$$match(L_h^c) = \underset{L_{\acute{h}}^{c+1}}{argmin} |L_h^c - L_{\acute{h}}^{c+1}|, h = 1, ..., l \tag{2.15}$$

$$match(L_{\acute{h}}^{c+1}) = \underset{L_h^c}{argmin} |L_{\acute{h}}^{c+1} - L_h^c|, \acute{h} = 1, ..., \acute{l} \tag{2.16}$$

where L_h^c and $L_{\acute{h}}^{c+1}$ are respectively the local maxima smoothed projection profile of slice $c - th$ and $(c + 1) - th$. For the second step, the separation seams are generated by minimizing the cumulative energy M:

$$M(i, 1) = E(i, 1) + min \tag{2.17}$$

$$M(i, j) = E(i, j) + \min(M_{i-1:i+1,j-1}) \tag{2.18}$$

where E is the energy map computed from the smoothed image I:

$$E(i, j) = |\frac{I_{i,j+1}^{\sigma} - I_{i,j-1}^{\sigma}}{2}| + |\frac{I_{i+1,j}^{\sigma} - I_{i-1,j}^{\sigma}}{2}| \tag{2.19}$$

In Alberti et al. (2019) the Labeling, Cutting, Grouping (LCG) method is proposed. The approach is based on a new energy map which is built from background energy and text energy. The **background energy** is computed as follows :

$$B(x_i) = \frac{1}{\underset{c \in CC}{min} ||l(x_1) - l(c)||} \tag{2.20}$$

where $l(.)$ denotes the coordinates of the pixel; CC denotes the centroid of all Connected Components in the image; $i = 1, ..., h \times w$.

The **text energy** is then computed as follows :

$$T(B(x_i)) = \begin{cases} B(x_i) & if \quad x_i = 1 \\ 0 & if \quad x_i = 0 \end{cases}$$

Then the final energy is the smoothed energy of text energy and background energy :

$$S(B(x_i), T(B(x_i))) = C_2(C_1(T(B(x_i) + B(x_i)))) \tag{2.21}$$

where C_2, C_1 are the convolution operations.

Then, the seed seams are cast every 100 pixels (vertical pixels) instead of using defined regions as in Arvanitopoulos and Süsstrunk (2014) and Asi et al. (2011). The final separation line is adjusted from the number of centroid groupings (number of centroids of connected components under the seam) (See Fig.2.8). A close approach to the seam carv-

Figure 2.8: Binning centroids under the seam (red lines) Alberti et al. (2019)

ing method is the A* path planning method Surinta et al. (2014). It consists of finding the separation lines that go from the starting state (s) to the goal state (g), by minimizing different cost functions. The local maxima represent the estimated text lines. Let's denote l_m is set local minima of the histogram projection profile then the position of the starting state ($s_m = (0, y_{l_m})$) and the goal state is set at the same height of s_m, g_m = (w, y_{l_m}). The cost function is the sum of two cost functions:

$$F(n) = G(n) + H(n) \tag{2.22}$$

where $H(n)$ is the heuristic cost function that estimates the distance from the state (n) to the goal state $((g))$.

$$H(n) = ||X_n - X_g|| \tag{2.23}$$

where X_n, X_g are the coordinates of the current state (n) (the position of the current pixel) and the goal state (g).

G(n) denotes the current traveling cost from starting state (s) to state (n). G(n) is the combination of two cost functions, detailed below, computed as follows :

$$G(n) = D(n) + N(n) \tag{2.24}$$

The first cost function is the minimum Distance Cost Function $D(n)$:

$$D(n) = \frac{C}{1 + min(d(n_{y_u}), d(n_{y_d}))} \tag{2.25}$$

where C is a constant value, $d(n_{y_u})$, $d(n_{y_d})$ are the distances from the current point to the closest text pixel computed in two ways: up and down (See Fig. 2.9). The second cost

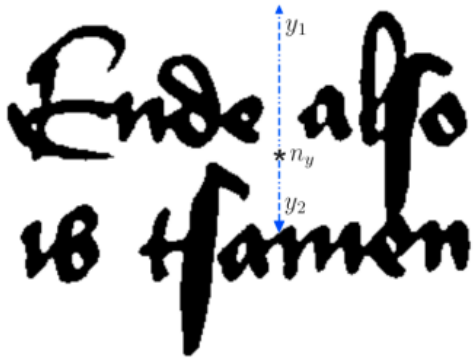


Figure 2.9: Illustration for the minimum distance cost function Surinta et al. (2014)

function is the Neighbor Cost Function $N(n)$:

$$N(n) = \begin{cases} 14 \quad if \quad NN(n) \text{ is in diagonal direction} \\ 10 \quad if \quad NN(n) \text{ is in horizontal/vertical direction} \end{cases}$$

where NN(n) is the neighbor node of current n. Figure 2.10 represents neighbor nodes of state (n).

- Bottom-up approach: In the bottom-up approaches Nicolaou and Gatos (2009) Cohen et al. (2014), Zahour et al. (2009), different topological features are used such as block, connected components, search trees, a d heuristic rules to extract the lines. Smearing method is one of the earliest methods which is used in many works Wong et al. (1982),

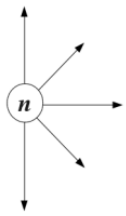


Figure 2.10: 5-directional movement of state (n)Surinta et al. (2014)

Kesiman et al. (2016a), Shi et al. (2009), Shi et al. (2005). The pioneer work is presented in Wong et al. (1982). A sequence of pixels is smeared through the horizontal (or vertical depending on the document structure) over a defined distance. The smearing image S of an input binary image $I \in R^{h \times w}$ can be calculated as:

$$S(x,y) = \sum_{i=-bw/2}^{i=bw/2} \sum_{j=-bh/2}^{j=bh/2} I(x+i, y+j)) \tag{2.26}$$

where bh and bw are the pre-defined blur distances relative to height and width. The connected components on the smeared image are considered as the boundaries of text lines. The typical smearing methods work with binary input documents, so in Shi et al. (2005), the authors introduce an Adaptive Local Connectivity Map (ALCM) which converts the gray input image to a new map space where higher values correspond to the text pixels.

$$A(x,y) = \int_R I(x,y).G_c(t-x,y).dt \tag{2.27}$$

where I(x,y) is the input gray image and

$$G_c(x,y) = \begin{cases} 1 & if \quad |x| < c \\ 0 & if \quad otherwise \end{cases}$$

where c is the size of the sliding window of the convolution. As the author recommended, a good initial c is equal to three times the average height of the text. Then, a thresholding method is applied to separate the text and background parts more clearly. Depending on the different rules defined by the shape and the distance between the components, the connected components can be removed if the rules are not satisfied or merged with

others to constitute the text lines. To solve the problems brought by fluctuating documents (shorter lines, irregular distance between lines), they extend the ALCM by using different filters such as steerable filter in Shi et al. (2009), isotropic Gaussian filter in Cohen et al. (2014). Instead of using only one filter, in Cohen et al. (2014), the smearing image L is created by a combination of smoothed versions of input image I as follows :

$$L(x, y, t_x, t_y) = g(x, y, t_x, t_y) * I(x, y) \tag{2.28}$$

where $g(x, y, t_x, t_y)$ is the anisotropic Gaussian filter

$$g(x, y, t_x, t_y) = \frac{1}{2\pi\sqrt{t_x t_y}} e^{-}(\frac{x^2}{2t_x} + \frac{y^2}{2t_y}) \tag{2.29}$$

Then, a component tree is used for the binarization of the smearing image. Figure 2.11 illustrates the process of binarization with a component tree. In the beginning, the root of

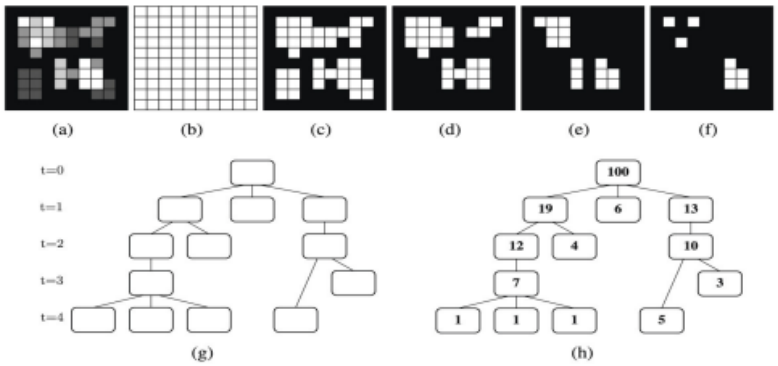


Figure 2.11: Illustration the process for selecting threshold of binarization

the tree is obtained at threshold 0. At this threshold, all the components in the smearing image are merged into one (threshold = 0, all pixels are white). The decomposition of the components is done through the increment of the threshold. The next node of the tree represents connected components generated by the new threshold. The process stops when the threshold reaches the maximum value of the original smearing image (all pixels are black). When the construction of the component tree is finished, a simple piece-wise linear approximation is applied to each node of the component tree to measure whether or not it is a text line. After this step, the validated connected components can be considered as the

text-line. To correctly label disconnected components or spurious lines, different functions are applied in a post-processing step. The final text-lines are obtained by minimizing the following objective function :

$$E(f) = \sum_{c \in C} D(c, l_c) + \sum_{c,\acute{c} \in N} d(c, \acute{c}).\delta(l_c \neq l_{\acute{c}}) + \sum_{l \in L} h_l \delta_l(f) \tag{2.30}$$

where $D(c, l_c)$ represents the cost of assigning c (connected components) to the label l_c (text-line), $d(c, \acute{c})$ represents the coherence between two text-line l_c and $l_{\acute{c}}$. h_l denotes the label cost for the appearance of each text line. The problem with this approach is when the line is fragmented, the smearing group can not be merged continuously. In addition, if the space between two consecutive lines is relatively small (ascending and descending characters touching the upper/lower line), this method fails.

- Deep learning based approach: Recently, deep learning-based methods for text line segmentation have emerged Barakat et al. (2020, 2021b); Grüning et al. (2019); Mechi et al. (2019); Renton et al. (2017); Vo et al. (2018a). Inspired by the semantic segmentation problem, different convolutional networks have been studied for the text-line segmentation problem such as Barakat et al. (2021b), Vo et al. (2018a), Renton et al. (2017). In Barakat et al. (2021b), the authors modified the fully convolutional networks Long et al. (2015) to adapt to the problem of text-line segmentation. The model is based on an encoder-decoder architecture (See Fig. 2.12). proposed The encoder model includes 5 convolution layers

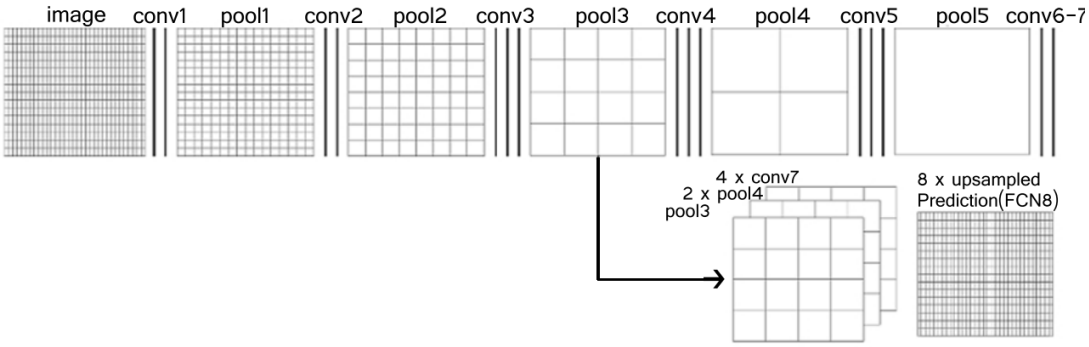


Figure 2.12: the FCN architecture proposed in Barakat et al. (2021b)

which are used to down-sample the input image. Decoder includes the transpose convo-

lution which up-samples to encoder output to return an image with the same size as the input image. Moreover, the input image is also adjusted to 320x320 instead of 224x224 as in the original work Long et al. (2015) to analyze the context around text-line. In Renton et al. (2017), a similar FCN is used but it differs from the work of Barakat et al. (2021b) at two points. The first point is that there is no resolution change of the input image in the whole encoder and decoder. The second point is the use of dilated convolution instead of transpose convolution or up-sampling. The reason for using dilated convolution is that text-line detection doesn't need a large context to be well segmented.

In Mechi et al. (2019), the authors adjust the original Unet (symmetrical encoder-decoder) Ronneberger et al. (2015) for text line segmentation. Specifically, each "down" step in the encoder is concatenated with the "up" step of the decoder (due to the symmetrical network) to provide more information when reconstructing the original shape of the input image (skip connection). In general, the results from the deep learning model usually appear as disconnected components (a line is separated into many components) or noisy components, the post-processing step with different heuristics cost function Barakat et al. (2021b) or line adjacency graphs Vo et al. (2018a) are used to improve the text line segmentation. In Grüning et al. (2019), a similar idea of combining deep learning techniques and image processing has been used. First, a deep convolution network namely ARU-Net is used to classify each pixel into different classes: baseline, separator, and others. A different strategy is used in the training model which allows to exploitation of input images at multi-scale levels and to sharing of weights between each scale. In the second step, since the results from step one contain too many separate pixels, the authors introduce the superpixels to represent the normal foreground pixels. They have to satisfy a distance condition with a set of initial pixels extracted by thresholding from a baseline map. For each superpixel, its text orientation θ and interline distance s is denoted as its *state*. Based on the state information of each superpixel, its label is assigned by minimizing the labeling cost function. Super pixels are clustered and merged to text-line under two assumptions: The baseline should not exceed a curvilinearity value (5 in the paper) and there is no baseline inside of the interline area. In Barakat et al. (2020), instead of using a supervised manner, the unsupervised approach has been used. The authors have trained Siamese network to

determine whether two patches of the image (extracted randomly from a list of cropped patches of the original image) are similar or different. The trained model is used as an embedding network to extract the blobs of the image. Energy minimization that combines different cost functions is used to split these blobs into the appropriate text lines. Overall, the disadvantage of the approach is the preparation of the training model because most of the available datasets are quite small and, most of the time, each training model needs a different ground truth format. Moreover, the training model is often adapted for the baseline which does not include the ascending and descending characters or the cases where the lines touch each other, a post-processing step is then needed.

2.6 Document Recognition

After segmenting the document components, the next step is the recognition task. Recognition is used to convert the component into a machine-readable format so that it can be used for later steps such as indexing, and retrieval. The recognition task is one of the main stages of the Document Image Analysis (DIA) pipeline. This task is one of fundamental work in the field of computer vision.

Two subcategories of this problem can be considered. Optical Character Recognition (OCR) typically used with printed documents where characters are generated by a defined/specific font. The second is Handwritten Character Recognition (HCR) used with manuscripts where the texts are handwritten, without specific rules, and whose style depends on the writers. The former achieves acceptable results on the popular language and is deployed in various commercial products Smith (2007) while the latter is still an open challenge for the community due to the variability of the characteristics such as writer's style, limited corpus and the quality of the documents. Based on the inputs of the traditional recognition systems and their evolution, we have roughly divided the study into three levels: page/paragraph recognition, line/sentence/word recognition, and glyph/character recognition.

2.6.1 Glyph/Character recognition

The lowest recognition level concerns glyph or character recognition. This problem can be considered as the well-known problem of classification. So research works usually follow the same workflow by using a feature extraction network combined with classifiers such as Support Vector Machine (SVM), Random Forests (RF), K-nearest neighbors (kNN), Decision Tree (DT), or template matching techniques. The feature extraction method is the most important step to achieve better recognition performance. So, many works focus on designing appropriate feature extractors for a specific language. Feature extraction methods can be divided into two main categories: handcrafted features and neural networks.

Handcrafted features

Handcrafted feature extraction involves techniques manually designed by domain experts. Numerous methods to extract handcrafted features have been proposed based on the characteristics of the studied language. An overview of different methods and their evolution can be seen in the literature Trier et al. (1996), Mori et al. (1992).

At first, the proposed method worked mainly with printed characters. Histogram projection profile has been proposed in Glauberman (1956). The problem with the projection profile is that the method is sensitive to font and size changes or noise. So invariant features have been proposed such as Gradient Based Shi et al. (2002), Neighborhood Pixels Weights (NPW) Kumar (2010), Zoning Bokser (1992), Geometric moment invariant Hu (1962), Kirsch Direction edges Kirsch (1971), Kim et al. (2004), Wen et al. (2007). In Shi et al. (2002), features are extracted by combining the gradient feature and the curvature feature. The gradient feature is represented by the direction and the strength of the gradient :

$$Direction : \theta(i,j) = tan^{-1}(\frac{\delta v}{\delta u}) \tag{2.31}$$

$$Strength : f(i,j) = \sqrt{(\delta u)^2 + (\delta v)^2} \tag{2.32}$$

where δu, δv are calculated from Roberts filter on the blurred input image.

$$\delta u = g(i+1, j+1) - g(i, j) \quad ; \quad \delta v = g(i+1, j) - g(i, j+1)$$

For the curvature of pixel x, the calculation is derived from Toriwaki and Fukumura (1978) and computed as follows :

$$R(x) = 1 - \frac{1}{2}\sum_{k \in S_2} f_k + \frac{3}{8}\sum_{k \in S_2} f_k f_{k+1} + \frac{1}{4}\sum_{k \in S_1} f_k \hat{f_{k+1}} f_{k+2} \tag{2.33}$$

where $S_1 = \{1; 3; 5; 7\}$; $S_2 = S_1 \cap \{2; 4; 6; 8\}$ are the 8 connectivities of pixel x. f_k is the value of pixel.

The zoning technique Bokser (1992) was motivated due to the limitations of the skeletonization method (ambiguity between the "8" and "s") and the limitations of topological and geometric characteristics (similarity between "A" and "R") Although these representations, in that case, are quite close, there are still differences at the level of small regions. Therefore, the input image is divided into small parts, then the features of each part such as the percentage of black pixels in the case of binary image or the average gray intensity are computed as the characteristic values of this part. Finally, each character/glyph is represented by a vector that concatenates the features of each part.

Neighborhood Pixels Weights (NPW), the idea of this technique is that pixels that are spatially close to each other are likely to be related in some way. Therefore, the value of each pixel is re-calculated by weighting its neighboring pixels (each weight is normalized by the maximum weight on that corner). In Kumar (2010), three neighborhood levels of size 8, 16, and 24 pixels are used to calculate the representative characteristic of the pixels. The weights of each neighborhood pixel are based on the relative position of the current pixel.

Kirsch Directional edge detection Kirsch (1971) is a non-linear edge detector that finds the maximum edge strength of an image in the directions: North (N), North West(NW), West(W), South(S), South West(SW), Sound East (SE), East (E), North Edge (NE). The computation can

be denoted as follows :

$$G(x,y) = \underset{z=1,..,8}{max} \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} g_{ij}^{z}.I(x+i,y+j) \tag{2.34}$$

where g_{ij}^{z} represents the set of 45 degree rotations of the kernel from $g^{(1)}$ defined as follows:

$$g^{(1)} = \begin{bmatrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{2.35}$$

the kernel $g^{(2)}$ corresponds to a rotation of 45 degrees of $g^{(1)}$:

$$g^{(2)} = \begin{bmatrix} +5 & +5 & -3 \\ +5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{2.36}$$

and so on for the other kernels. For example, in Kim et al. (2004) from the input 16×16 images, 4 gray edge images (N, S, E, W) computed from Kirsch edge detection are binarized with a threshold of 10. Then, these directional images are compressed into small regions of size 4x4. For each region, feature maps are computed by accumulating the pixel values.

An extensive experiment, with the different handcrafted features detailed above, conducted on Baseline manuscripts can be found in Kesiman et al. (2016b). At the beginning of our project, some efforts have been made to recognize the glyphs of the inscriptions Nguyen et al. (2019b), Nguyen et al. (2019a). A dataset consisting of 1607 images with 37 glyphs was created. In Nguyen et al. (2019b), Study different methods of feature extraction such as Histogram of Gradient (HOG), Neighborhood Pixels Weights (NPW), and GoogleNet with different classifiers such as: kNN and Support Vector Machine (SVM) have been done on this dataset. A better result was obtained by using GoogleNet feature extractor and SVM classifier. Experiments combining data augmentation and transfer learning have been carried out to further improve the results Nguyen et al. (2019a).

Neural networks

The handcrafted features require knowledge of the language being studied and may not generalize well to other languages, this weakness can be handled by neural networks with their hyperparameters. Various architectures have been proposed and evaluated for the character recognition task. In the early work, Multi Layer Perception (MLP) Shamsher et al. (2007) applied on Urdu script. The input image (size 10x15) is flattened into a vector of size 150 and then put to the three fully connected layers of dimension 150/250/16, respectively. Recently, the availability of large labeled datasets and training resources has led to a huge improvement in the classification performances with neural networks. The CNNs architecture such as Lenet LeCun et al. (1998), VGG Simonyan and Zisserman (2014), Resnet He et al. (2015), Tan and Le (2019) has been studied both printed and handwritten dataset Liu et al. (2003), Nguyen et al. (2017), Ciresan et al. (2011), Husnain et al. (2019). In Khurshid et al. (2009), the proposed architecture consists of two convolutional layers (each convolutional layer includes convolution layer, pooling, and activation function) with two fully connected layers. This architecture is used to recognize Urdu characters. In Ciresan et al. (2011), different configurations of CNNs have been proposed for NIST digits and NIST letters [1] and achieved results close to the human performance.

2.6.2 Line/Sentence/Word recognition

When recognition is applied at the glyph/character level, recognition is done on a character-by-character basis. At the higher level, where the character may be affected by its neighborhood or nearby words, other techniques that take this problem into account must be applied. Generally, the existing approaches can be roughly divided into language-free and language-based approaches.

Language-free approaches

In the first type of approaches Shi et al. (2015), Baek et al. (2019), Borisyuk et al. (2018) the general flow (See Fig. 2.13) is based on the use of a Convolutional Neural Networks (CNNs) to extract the high-level visual features, then a Recurrent Neural Networks (RNN) to make pre-

[1] https://www.nist.gov/srd/nist-special-database-19

dictions which are guided through a Connectionist Temporal Classification (CTC) Loss Graves et al. (2006). One of the pioneer works is the Convolutional Recurrent Neural Network (CRNN)

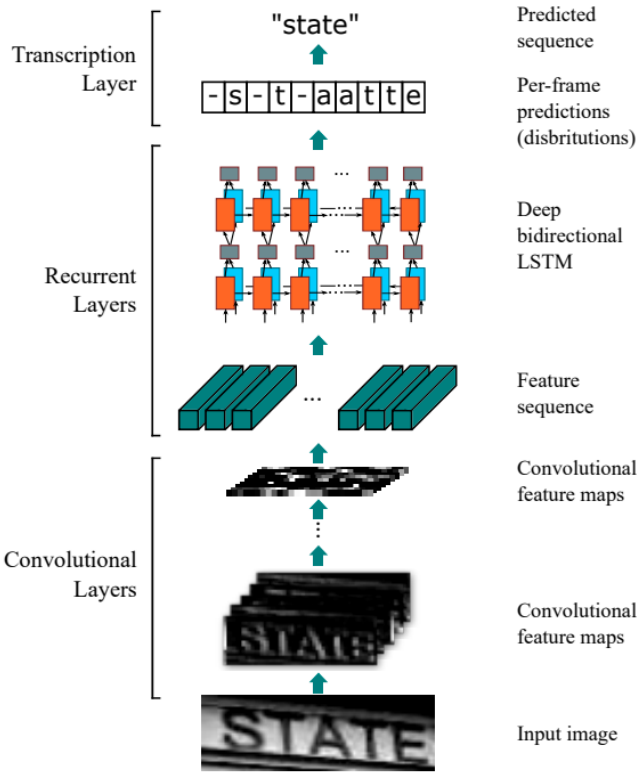


Figure 2.13: Illustration of a Convolutional Recurrent Neural Network (CRNN) Shi et al. (2015)

model proposed in Shi et al. (2015). The CRNN model consists of three parts: Convolutional layers, Recurrent layers then Transcription layers (See Fig. 2.13). The Convolutional layers are the combination of multiple standard blocks: convolution layer, pooling layer, and an activation function. Recurrent layers are the stack of Recurrent Neural Networks (RNN) which is proposed to learn the relation between the sequence features from convolutional layers. The Transcription layers are fully connected layers that make predictions for each output feature from recurrent layers. Since the output prediction can be repeated (in case of characters separated by a responsive field) or empty (in case of an empty receptive field), the CTC Loss will be used to align and train the model. In Borisyuk et al. (2018), the authors use Resnet architecture He et al. (2015) for feature extraction then CTC Loss as the objective function for training the recognition with scene images. To consider documents with inclined lines, they propose a module to transform the image into a straight line before using the normal process Shi et al. (2018), Cheng et al.

(2018). The disadvantage of this approach is that when the quality of the input image is low due to the presence of noises, and unclear, and occluded characters, they disturb the model, leading to incorrect predictions.

Language-based approaches

In the second type of approach, a language model is used to provide additional linguistic information. This language model can be integrated into the training phase Nguyen et al. (2021a), Kang et al. (2019), or implicitly from the learning of training data. The recognition task can, then, be viewed as a sequence-to-sequence problem in Nature Language Processing (NLP). Therefore, many architectures are proposed inspired by similar solutions in NLP. The general method consists of two parts: an encoder and a decoder. The encoder part consists of a CNNs to extract the features. The decoder part consists of a RNN/transformer which proposes to implicitly/explicitly learn the language model. To enhance the ability to capture relevant information from the input sequence while generating the output sequence, an attention-based technique has been proposed Kang et al. (2018), Kass and Vats (2022), Michael et al. (2019). The principle of using an attention module between encoder and decoder is shown in figure 2.14. In detail,

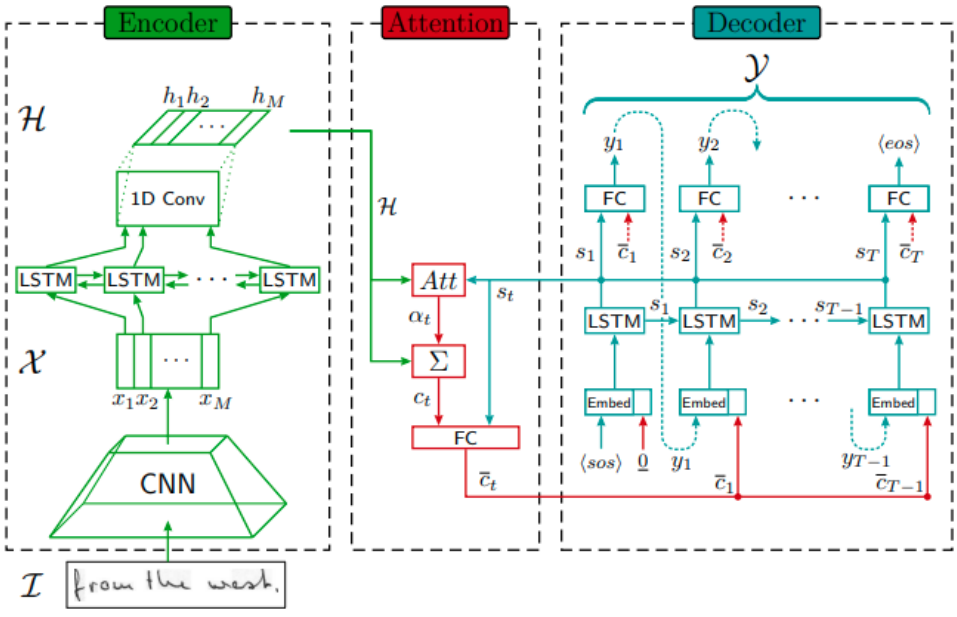


Figure 2.14: Attention-based sequence to sequence Michael et al. (2019)

at each time step during the decoding phase, the model prediction t is based on the previous

character $t-1$ and the new context vector $\acute{c}_t$ which is weighted by the features $\mathcal{H}$ of the hidden-state feature extractor network s_{t-1} of the decoder network. Instead of using separately these visual information and linguistic information, many works propose interactive and multi-modal approaches to improve the representation of the features Na et al. (2022), Aberdam et al. (2022), Fang et al. (2021). The interaction between the two modalities brings a better representation of the features for the final prediction. In Baek et al. (2019), the authors generalize the solution for the OCR problem into 4 modules: Transformation, Feature extraction, Sequence Modeling, and Prediction (See Fig. 2.15). For each module, different backbone architectures were used

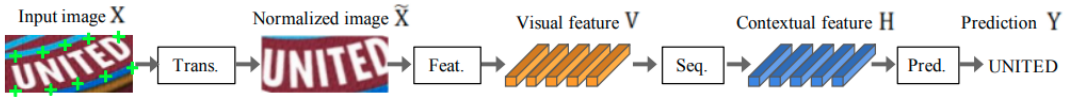


Figure 2.15: Proposed 4 modules for OCR problem Baek et al. (2019)

to compare regular and irregular datasets. Various aspects such as performance, time inference, and number of parameters have been analyzed to determine the trade-off between them. The best accuracy model is the combination of thin-plate spline (TPS) transformation, Resnet, Bidirectional Long-Short-Term-Memory (BiLSTM), and Attention-based sequence prediction.

Knowledge distillation is also an interesting approach by leveraging the knowledge from a huge dataset and transferring it to a new domain. More precisely, the different modules of standard OCR (feature extractor or sequence model) are trained separately with different objectives. Then, the whole model built from separated modules is trained in an end-to-end manner by the standard OCR objective function Vogler et al. (2021), Lyu et al. (2022), Li et al. (2021). For example, in Vogler et al. (2021), taking the idea of the masked language model in Vaswani et al. (2017), the authors first pre-trained the feature extractor with a masked strategy. A random part of the input image is hidden, the model has to predict the missing part through the surrounding area information. In this way, the model can also learn linguistic information from the image. Then, the trained model is added to the recurrent network module to make the sequence prediction. Authors show that with few fine-tuning samples (30 text lines), the model can converge. In the second scenario, the standard OCR model is trained on a huge corpus, Then, fine-tuning is directly applied to the trained model with specific domain data and a small set of ground truth

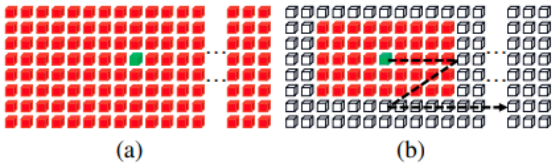


Figure 2.16: Illustration of (a) global mixing and (b) local mixing Du et al. (2022)

data.

Most of the works mentioned above have to use RNN which requires a sequential process during the decoding phase, some works try to avoid this phenomenon by using only a CNNs and a MLP Atienza (2021), Kang et al. (2022), Yousef et al. (2018),Coquenet et al. (2020), Du et al. (2022). In this approach, the model often needs to study by itself the linguistic information from the input image. In Atienza (2021), inspired by the effectiveness of vision transformer Dosovitskiy et al. (2020), the authors adapt it for recognition by modifying only the output head (number of nodes of output is the number of characters). The results are quite competitive compared to methods using recurrent networks but faster in inference time. In Du et al. (2022), the authors propose two new modules: global mixing and local mixing (See Fig. 2.16). The idea of the two modules is to capture both local and global dependency information through the self-attention mechanism. While global mixing learns the dependence among all character components, local mixing learns the local relationship inside the pre-defined window. If the prediction is done in one step, in some cases, the model can make mistakes. In order to improve the predictions, an iterative prediction process has been proposed Qiao et al. (2021), Fang et al. (2021), Na et al. (2022), Bhunia et al. (2021). The main idea of this strategy is to use an iterative mechanism in the decoder combined with visual information to enhance the prediction through each step.

2.6.3 Page/Paragraph recognition

The latest study in this field can be done at paragraph or page level Bluche (2016); Bluche et al. (2017); Coquenet et al. (2022); Ly et al. (2019); Singh and Karayev (2021b); Wigington et al. (2018); Yousef and Bishop (2020). The previous approaches (at line or sentences level) assume that the text is on the same lines, so the sequence prediction will be generated across the width of the image. However, input as a page or a paragraph consists of multiple lines,

so the typical approaches can not properly work. Therefore, the modification of the training process or the combination of segmentation has been proposed to train the whole model. One of the basic ideas is the work of Yousef and Bishop (2020) that considers the whole paragraph as one long line (merging the multiple lines of the ground truth into a unique line) so that the model can be trained with a standard CTC Loss (See Fig. 2.17). The drawback of this method is the definition of some parameters that depend on the dataset such as the length of the line (maximum characters in the whole paragraph) and transpose the shape in the decoder network (to make sure to have enough prediction). To avoid pre-estimating the text lines, some works

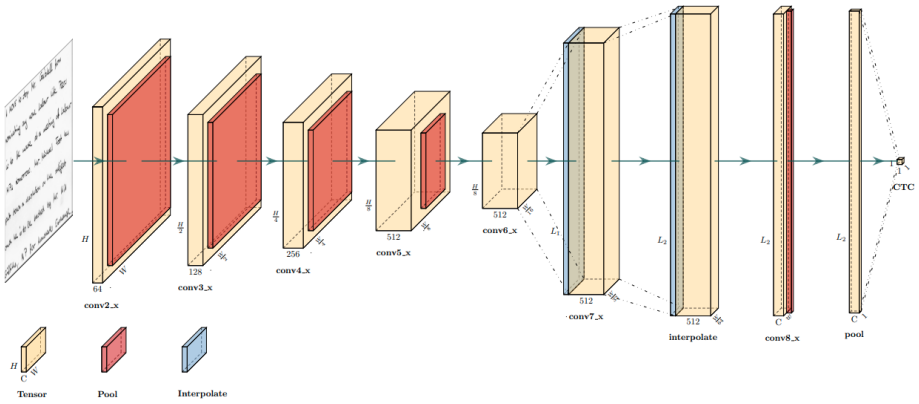


Figure 2.17: The proposed architecture OrigamiNet Yousef and Bishop (2020)

try to combine segmentation and recognition. Using this strategy, errors can be reduced during segmentation works and also improve recognition performance. The first combination of the two tasks is proposed in Bluche (2016). In this work, the authors changed the standard collapse layer to a weighted collapse layer to help the model focus on the relevant parts of the image feature. Another idea consists of using the implicitly learned line segmentation task while doing the recognition task Coquenet et al. (2022), Singh and Karayev (2021b) Ly et al. (2019). In the Ly et al. (2019), the attention mechanism is used to assign different weights to the different parts of the image by taking into account the spatial relationships. Unlike Bluche (2016); Ly et al. (2019), in Coquenet et al. (2022), during the first training phase, text line images are used as a normal training model, then the authors used a vertical weighted mask to determine the relevant part of each line. The training objective function is combined with the recognition loss and the text line loss. Instead of an implicit segmentation of line, in the work of Singh and Karayev (2021b), the authors use a sliding window over the whole image. Each patch (that contains a

word or characters) is predicted and then concatenated to the final prediction of the entire page.

2.6.4 Transliteration

The main principle of a character recognition system consists of a mapping between the image of one character and its textual representation. The output results are then readable and understandable. However in Cham language, studied in this work, some special phonological rules are used. The meaning of a character can depend on the previous or the next character. The mapping is therefore more complicated. The system must therefore have the capacity to learn these implicit rules. This conversion is called transliteration. Transliteration involves matching phonemes from a source writing system to a target writing system.

Transliteration is not a task very studied in the literature, so research works applying these new approaches are still limited. There are two kinds of problems related to the transliteration. The first concerns text-to-text transliteration meaning that a text source of one script is transcribed into a text source of another script. The second is image-to-text transliteration, that is, from an image of written text, the system must extract and recognize the text of a script before transcribing it into a target script. Most of the prior work AbdulJaleel and Larkey (2003), Shishtla et al. (2009), Finch and Sumita (2010), Arbabi et al. (1994), Fujii and Ishikawa (2001) related to text-to-text transliteration often used techniques based on statistical model. For example, in Shishtla et al. (2009), the transliteration algorithm is split into two phases. In the first phase, characters are aligned then in the second phase some statistics are applied to generate the target language. In Finch and Sumita (2010), the authors propose a two-step approach by using the joint multi-gram model to generate a set of candidates for the transliteration and then rescores each candidate using the phrase-based statistical machine translation (SMT) system. In Arbabi et al. (1994), a hybrid algorithm that combines the knowledge-based system (to construct the vowelization rules) and artificial neural networks (to filter the input names that can not be vowelized) has been proposed to generate multiple English spellings for Arabic person names. Recently, in the thesis of Kesiman Kesiman (2018), the author proposes two ways to transliterate Balinese script to Latin script by using segmentation-based and segmentation-free based. In the first scheme (text to text), based on the segmentation of glyphs, phonological rules with knowl-

edge representation are used for automatic transliteration. In the second scheme (image to text), different configurations of hyper-parameters of architectures based on RNN-LSTM networks of OCRopy [2] have been studied for word transliteration but the performances are quite low.

2.7 Off-the-shelf system for document analysis

With the development of a scientific community working on historical documents, the results are gradually improving by adapting different techniques from other fields to this problem. To be easily accessible to a large number of people, some software has been created and built on a unified block. This software can thus be provided to end users with no technical knowledge. Here we introduce two popular software for DIA.

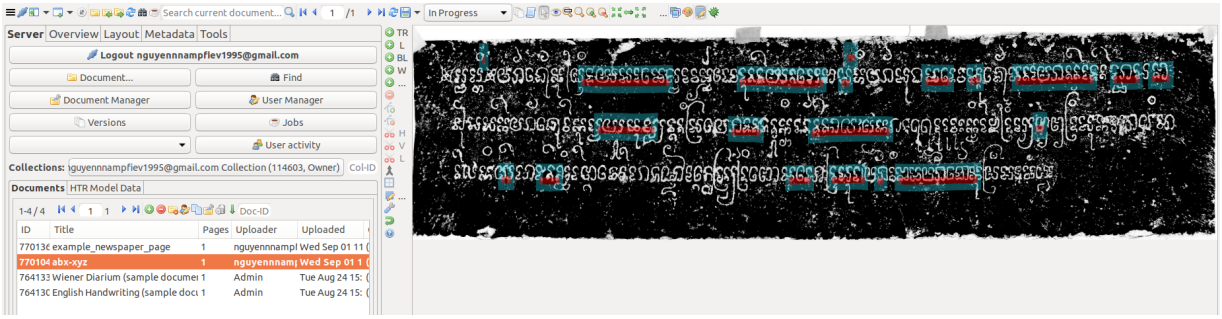


Figure 2.18: Transkribus Application

Transkribus Kahle et al. (2017) is one of the first interactive computer-assisted platforms for the transcription, recognition, and retrieval of digitized historical documents. The two main tasks supported by this tool are layout analysis and text recognition. To start with this tool, users can upload a set of documents and prepare the ground truth corresponding to each task. The training process can be done by selecting pre-trained models that are embedded in the back-end then adjusting the hyperparameters as well as splitting the dataset. One of the advantages of this tool is that it provides an online server to perform the training process thus reducing the problem of access to training resources.

Based on the work of Transkribus Kahle et al. (2017), eScriptorium Kiessling et al. (2019) was

[2] https://github.com/tmbdev/ocropy

built on the work of Gruuening et al. (2017) for the document layout analysis and Kraken [3] for the recognition task. The most important improvement in comparison with Transkribus is the user-definable architecture where user can define their models instead of only using the models provided by the tool, making the tool more powerful. Moreover, eScriptorium is also more user-friendly. The interactions are easier and more intuitive. The tool offers more functionalities such as region visualization, adjustment of the bounding boxes, ground truth modification, and import and export of the models.

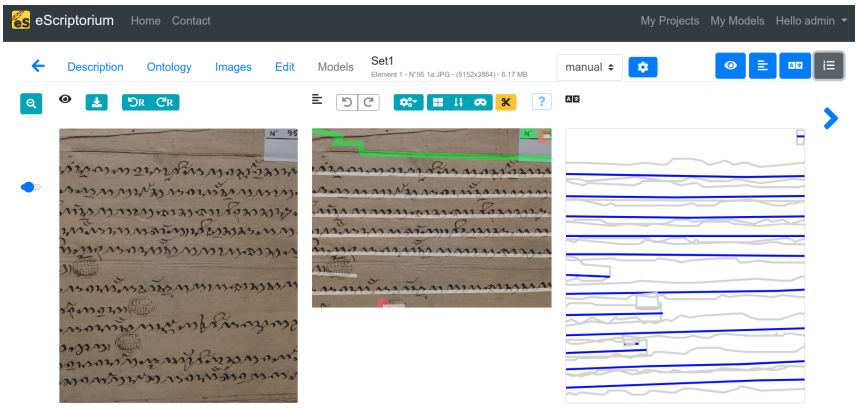


Figure 2.19: eScriptorium Application

With both tools, people can share as well as use many pre-trained models in different languages. This is an important point to increase the attention of the community working on historical documents, especially for researchers in human sciences whose technical knowledge of artificial intelligence is limited.

2.8 Conclusions

In this chapter, we review the general tasks applied in DIA. These tasks can be roughly split into four categories: digitization, pre-processing, analysis, and recognition. Depending on the properties of the input document and the final output requirements, different tasks have been proposed by researchers. Although the tasks mentioned above have been studied for a long time and obtained competitive results, there are still challenges for the research community working

[3] https://kraken.re/main/index.html

on historical documents. Indeed, most of the proposed methods are adapted to the characteristics of the studied languages and generalization is always difficult.

In the framework of the CHAMDOC documents studied in this thesis, the the whole pipeline follows the typical document image analysis workflow which includes: image pre-processing, text line segmentation, and text line transliteration. Although many pre-processing techniques and line segmentation methods have been developed, their application to a new script like Cham has provided mixed results. Transliteration, on the other hand, is still an open challenge because less work has been studied on this problem. To analyze the content of Cham manuscripts and inscriptions, our work has focused on three main tasks of the DIA pipeline: image denoising, text line segmentation, and transliteration. The works found in the literature served as a basis for developing new approaches allowing the analysis of Cham documents. The proposed methods are detailed in the following chapters. However, to evaluate the relevance of these methods, the creation of datasets was fundamental. The following chapter presents the characteristics of the Cham language: its history, its evolution over time, and the corpus used to build useful data sets.

CHAPTER 3

Collection of Champa documents

3.1 Champa Communities and Languages

3.1.1 History of Champa

So, let's start with the spelling of its name. The words "Cam" and "Campa" are written in Cham using the letter c- and not ch-. The habit of writing "Cham" or "Champa" was adopted from the English and Vietnamese languages for the convenience of pronunciation. Thus, although we are fully aware that it is preferable to follow the vernacular script, we will follow English conventions here. The Champa kingdoms were formed in the lower valleys of the rivers opening on the East Sea (South China Sea), between the provinces of Quang Binh in the North, and Binh Thuan in the South. The chiefdoms then gathered into the Champa kingdoms, and flourished from the end of the 2^{nd} century AD (See Fig. 3.1).

[1] https://adventurejourney.vn/vietnam-travel-blog/champa-kingdom-facts-with-history-and-map-territory.html

Figure 3.1: Champa's territory (Yellow parts)[1]

First of all, we know that they settled next to the Sa Huynh sites and that they have developed warehouses for trading products (rare woods, elephant tusks, rhinoceros horns, etc.) mainly taken from the surrounding highlands among indigenous populations (Katu, Edé, Raglai, Jarai, etc.). Some sculptures, found in this ancient period, are considered testimonies of religious worship. From the 6^{th} - 7^{th} centuries onwards, kingdoms were structured around self-proclaimed royal figures, through means of expression strongly influenced by India (including religions from India, Hinduism and Buddhism). The first brick temples appeared, as well as the first inscriptions, written in Sanskrit with a script copied from the Indian Brahmi.

Gradually, the kingdoms occupied all the lower valleys, grew in size and wealth, and raised dissensions with their neighbors (Java in the South, Khmer country or the peoples of the high plateaus in the West, and Viet authorities in the North from 11^{th} century). The Champa kingdoms experienced a phase of extreme wealth between the 9^{th} and 11^{th} centuries. They relate the influence of their kingdoms in inscriptions written in Sanskrit and ancient Cham and mark their territories with magnificent brick temples. They struggled to maintain their economic power on the maritime Silk route in the South China Sea by competing with each other to attract foreign merchants and by starting territory wars with neighboring countries.

They began to weaken in the 12^{th} century due to endless wars. They regained stability in the

13^{th} century by allying with the Dai Viet against the Mongol invasions. They almost regained their supremacy in Central Vietnam in the 14^{th} century but in a territory reduced to the area of Panduranga in South Vietnam. After 1471, they abandoned part of their former territories to the Viet power but retained their former southern kingdoms, called Panduranga. However, the impact of the Viet culture remained barely perceptible until the 16^{th} century. The last Champa kingdom only disappeared after the complete establishment of the Nguyen dynasty in the Southern part of present-day Vietnam. From the end of the 17^{th} century, with the management of Viet authority, the local Champa authorities had to adapt their entire administration. Manuscripts written in *Middle Cham* at this time can be seen as evidence of the complicated cohabitation between the administrations and the Champa and Viet populations.

Over time, the Champa people lost their rights, territories and autonomy. In 1832, the last Champa kingdoms disappeared politically. Champa populations continue to live in present-day Vietnam, mainly in the provinces of Ninh Thuan and Binh Thuan, but the other diaspora in Cambodia and Malaysia today constitute the majority of the populations still speaking *Modern Cham*.

In general, through many pressures within the neighborhoods, the Champa civilization was able to maintain their distinct identity and culture, and its legacy continues to influence the region today. Right now, to further explore the information of the kingdoms, there are a few resources that have been slowly damaged over time:

- There are still the remains of some historical sites such as temple towers and ruins (Po Nagar, Po Klong Garai, Po Rome, Po Sha Inu), stone sculptures (which can be seen today in Vietnam museums, mainly in Hanoi, Da Nang and Ho Chi Minh City) (See figure 3.2).
- The inscriptions mainly written in *Old Cham* language or *Sanskrit* over stone surfaces (See figure 3.3).
- Historical books, manuscripts, diplomatic and administrative documents written in *Middle* or *Modern Cham*.

The few resources available and the lack of transmission of knowledge lead to the disappearance

of the Cham language.

My Son temple | Stone sculpture

Figure 3.2: My Son temple and Stone sculpture

Inscription C. 233 written in Sanskrit | Inscription C. 42 written in Old Cham

Figure 3.3: Samples inscription written in Sanskrit and Old Cham

3.1.2 Languages

It is difficult to determine the correct form of the Cham language dating back to the early Champa era due to the lack of records from that time. Cham language is similar to the Proto-Malayo-Polynesisan languages Grant, Sidwell, et al. (2005). Today, Cham language has two variants called: Western and Eastern Cham Van Han et al. (1997). The Western Cham is used by people in Cambodia while the Eastern Cham is used mainly in Vietnam. Approximately, Champa peo-

ple speaking/reading *Modern Cham* language all over the world are about 800.000 peoples.

Cham scripts originate from the Brahmi script, coming from India Epigraphy (1998). This writing system was adjusted to the phonological contingencies of this Austronesian language. Each cultural sphere in Southeast Asia has sorted out and developed particular characteristics of the late Southern Brahmi prototype called the Pallava. The writing order is often from left to right. The peculiarity of the original writing is an abugida or alpha-syllabic system. Each letter is an abugida, which means that the symbol transcribes an open syllable consisting of a consonant followed by its inherent vowel *-a*. The other vowels are transcribed utilizing graphic symbols noted above, beside or below the graphic symbol of the consonant. Figure 3.4 shows the combination of a consonant with different vowels with its transcription in a context of transliteration.

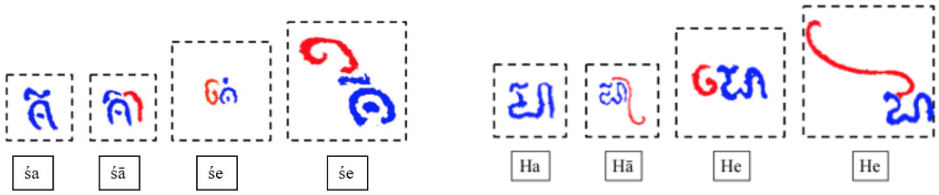


Figure 3.4: Illustration of Brahmi script system combining a consonant and other vowels to represent syllables.

In general, the evolution of Cham alphabet has undergone significant changes and has formed three types: *Old Cham*, *Middle Cham* and *Modern Cham*.

- *Old Cham* alphabet consists of 31 consonants and 11 vowels/diphthongs dating from 6^{th} to 15^{th} century.
- *Middle Cham* alphabet (figure 3.5) is an extension of the *Old Cham* consisting of 33 consonants and 26 vowels/diphthongs dating from 16^{th} to 18^{th} century.
- After the 19^{th} century two different dialects were formed: *Western Cham* (using in Cambodia) and *Eastern Cham* (using in Vietnam). Both come from Old and *middle Cham*,

but include strong influences from Cambodian and Vietnamese languages that make them hard to understand when taught to other people.

Today, people using these two dialects (*Western Cham* and *Eastern Cham*) no longer understand each other, because lexical borrowings were made in their country of origin (Cambodia or Vietnam) and have nothing in common apart from the basic grammar, even the scripts have become different.

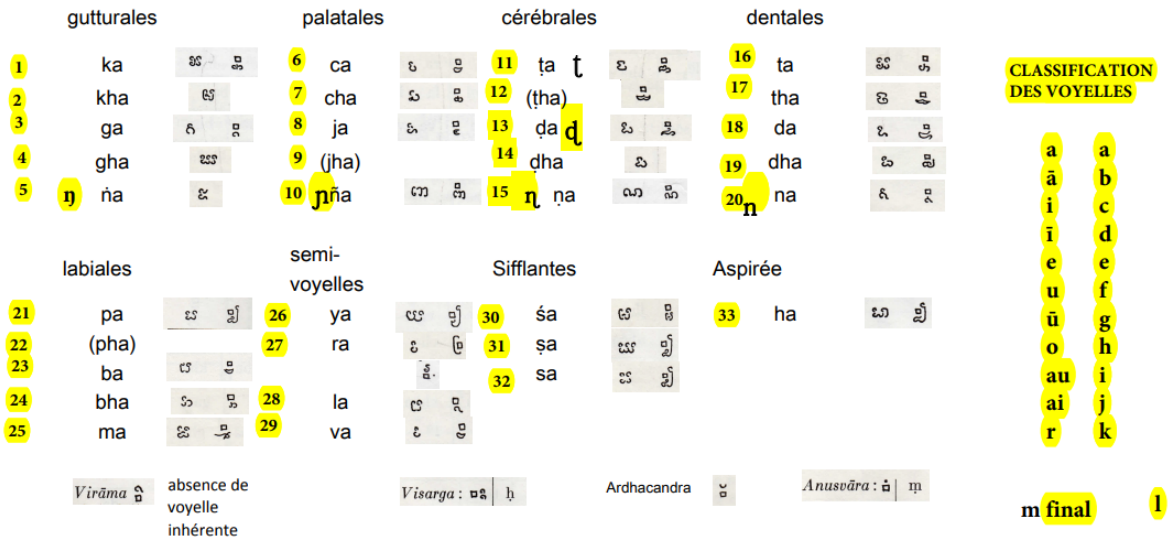


Figure 3.5: *Middle Cham* alphabet

3.2 Inscriptions

The first collection used in this work is called *Cham inscriptions*. In total, this collection includes around 200 inscriptions mainly found in the North Central and South Central Coast of Vietnam. The inscriptions that have been collected fall within a wide chronological range from the 6^{th} to the 15^{th} century A.D and are written either in Cham or in Sanskrit. The Sanskrit language is found between the 6^{th} and 10^{th} centuries, while the Cham language is used between the 11^{th} and 15^{th} centuries.

The script of Cham inscriptions is a writing system derived from the Indian Brahmi. The Brahmi writing system is the ancestor of Cham script, as are many Southeast Asian writings. The use of these two languages leads to a slightly different writing system, which must be considered for

the analysis of the text. Most of the information present in these inscriptions concerns: religious donations, and praises to the gods, kings, and their family members.

The images of Cham inscriptions have been obtained by a stamping process. (See Fig. 3.6). This digitization has been done in Vietnam by archaeologists during field excavations or by curators in museums when the steles were deposited there. The stamping process allows to copy the inscriptions carved on the stone (old stele) onto a large sheet of paper. The sheets of paper are then scanned to obtain digital images. Figure 3.8 shows the evolution of two Cham

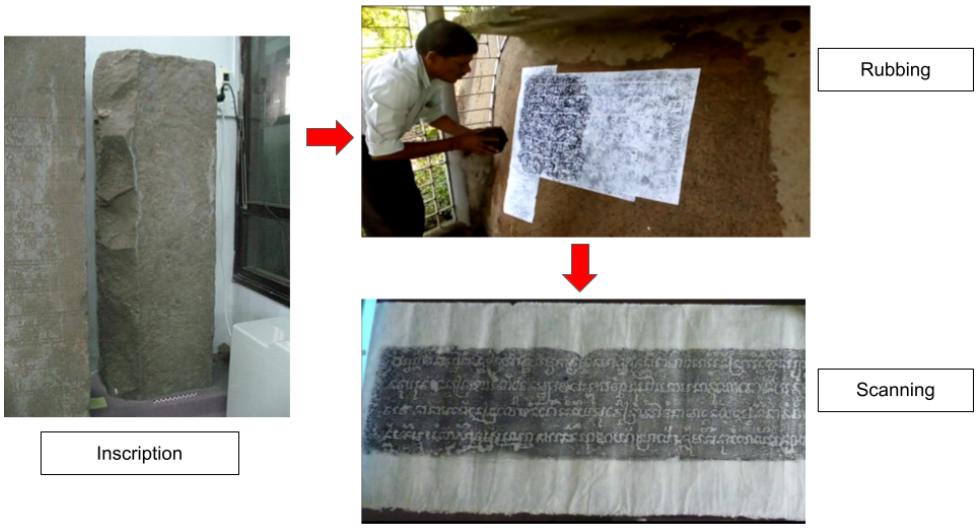


Figure 3.6: Illustration of the stamping process

characters over time. In the early, Cham letters were initially close to the Pallava script Phuong and Lockhart (2011). Over time, there are more consonants and vowels, some letters become simpler but others seem more complex with more strokes and curves (See Fig. 3.7).

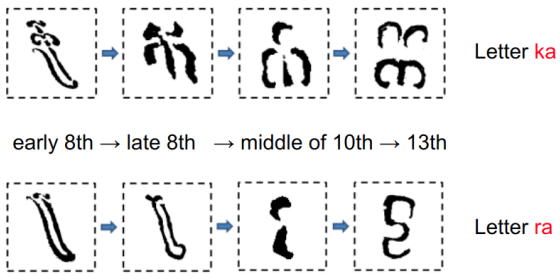


Figure 3.7: Evolution of letters ka and ra Nguyen et al. (2019b)

Figure 3.8: Evolution of Cham letters from 6^{th} to 15^{th} century in the inscription collection

3.3 Manuscripts

The second collection used in this work is called *Cham manuscripts* (See Fig. 3.9). These

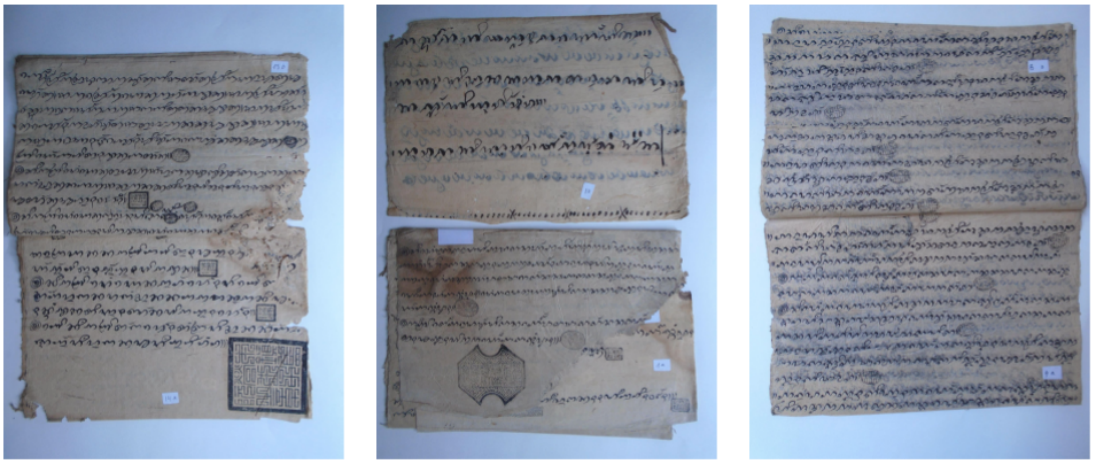


Figure 3.9: Sample Cham manuscript documents

manuscripts were collected from documents belonging to the Société Asiatique and are kept at the Collège de Paris.This collection, called the **Royal Archives of the Panduranga**, includes 551 manuscripts, and nearly 11,000 pages written on "Chinese" paper, which had never been published yet. The corpus of manuscripts is written in two languages and scripts: *Han Nom* (about 5% of the corpus) and *Middle Cham*. A large part of the manuscripts can be dated (from Vietnamese seals) between 1702 and 1810, making this set of manuscripts a coherent corpus of 18^{th} century texts. The manuscripts present the Asian society come from a royal deposit of a Cham lineage: that of King Po Klong Manai (1622 to 1627), whose temple to the ancestors (*kut*) is in the province of Binh Thuan. These manuscripts are the legal proof of Champa possessions

and their management. The inventory of this royal deposit was drawn up at the beginning of the 20^{th} century by Henri Parmentier and Father Eugène-Marie Durand. It was published in 1905 in the Bulletin of the *Ecole Française d'Extrême-Orient*. The manuscripts would have been brought back to Paris by Father Durand in 1906. These archives mainly consist of lists and declarations of rice fields, receipts, contracts, orders, petitions, reports, and judgments.
The obvious lack of study of these manuscripts is explained by the poor condition of the documents which are often damaged, sometimes torn or burned. The lists, requests, and judgments that compose them are data that were not intended to convey a political message, as is the case with other (dated) documents such as Cham inscriptions or Vietnamese annals. These documents relate the daily life of the inhabitants of ancient Panduranga during a relatively unknown period and complete our perception of the complex process, also little known, of the annexation of the Panduranga by the Nguyen dynasty.
As it evolved and with the decreasing use of Sanskrit, the Cham alphabet probably improved by adapting to everyday pronunciation. The Cham alphabet has also been found to have changed over the 16^{th} and 17^{th} centuries, but linguistics researchers have no idea, at this time, of the process that caused this transformation. As with *Old Cham*, the transliteration uses the open syllable system. Figure 3.10 illustrates some combination of the consonant k with other vowels and diphthongs.

3.4 Ground truth construction

Since only the digitization of each collection was available, the first work was to prepare the ground truth to build annotated datasets. Moreover, to evaluate each step of the pipeline in Document Image Analysis (DIA), different types of data have been processed.

This work was done from scratch thanks to the collaboration between our linguistic expert and our team in computer science. The main difficulty we face when preparing the data concerns the human resources available. Unlike other languages, where many people can understand them, few people can read the old inscriptions in *Old Cham*. There are only 5 experts in the world able to read it. So, a huge work had to be done manually by our expert.

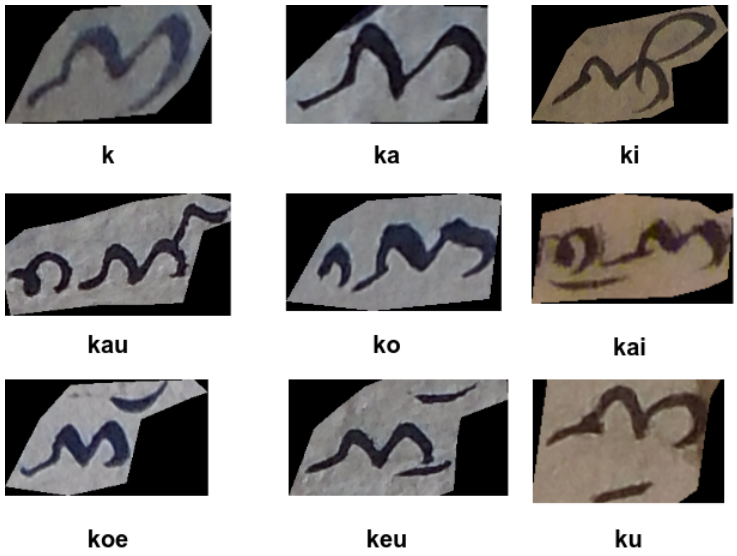


Figure 3.10: Some combinations of the consonant k with other vowels or diphthongs in the 18^{th} century in manuscripts.

All the processes for preparing the ground truth are supported through Fiji software Rasband (2011). Fiji is an open-source software that can be used for many tasks of image analysis. One of the advantageous features of Fiji is the support for plugins and macros that enhance its functionality. The interface of the application can be seen in the figure 3.11.

Figure 3.11: Preparation with Fiji

The workflow to generate the ground truth for each Cham inscription document is as follows:

- Due to the high degradation of the documents, a pre-processing step for enhancing the quality of the documents is applied. The annotation was done manually by our linguistic expert. A raster graphic editor was used to remove all pixels of noise and correct the missing pixels of text. As the cleaning task is time-consuming and to avoid processing large-size documents (since the dimensions of input images are between 1000 and 6000 pixels) that are too heavy for the annotator, each "document" image has been split into text line images as shown in figure 3.12. This ground truth will be used to evaluate both image enhancement and text line segmentation tasks.

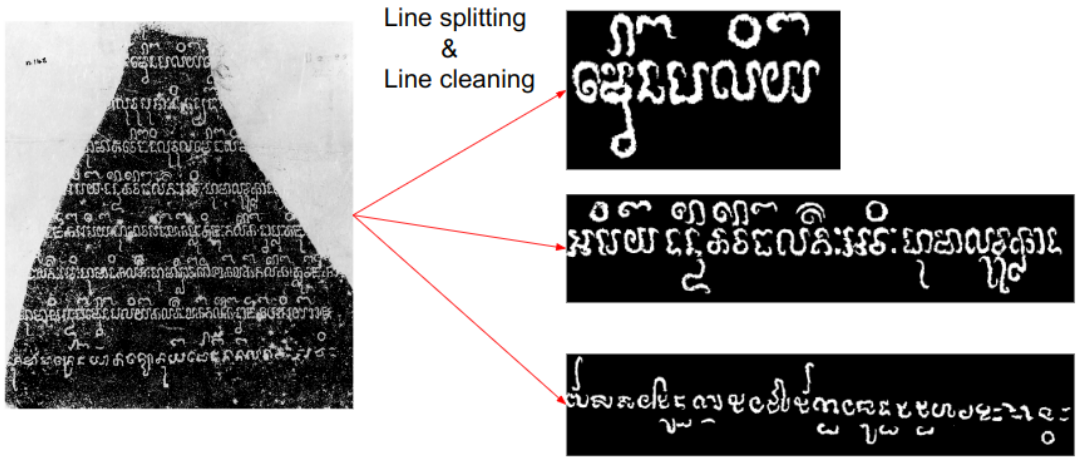


Figure 3.12: Line split and cleaned by our linguistic expert

  - For the image enhancement task, the dataset **Denoising-Cham** was constructed with 190 text lines from 21 inscriptions (written in Cham and Sanskrit script).
  - For the text line segmentation task, the dataset **Textline-Inscription-Cham** was constructed with 395 text lines from 26 inscription images.

- For each segmented text line above, the corresponding transliteration (in Latin) ground truth is prepared. Moreover, the writing script type (Sanskrit or Cham) is also provided. So far, the dataset includes around 700 text lines with transliteration from 42 inscription images (See figure 3.13).

For the Cham manuscript documents, we started the ground truth construction to evaluate the text line transliteration task. The annotated dataset includes text line images and the correspond-

C.119 A	Line 1	Sanskrit	⊙ svasti || śrīmān chrī parameśvaro narapatir nnānākalaikālayo dharmme
	Line 2	Sanskrit	ddho valavīryyakīrttiparamaḥ san sarvvabhaumaḥ kṛtī jitvā Pāpaka Pāṇḍuraṅganṛgaṇān hastādrirandhre śake senā
	Line 3	Sanskrit and Cham	sāṅkhyakṛtau śilācayam imaṃ saṅthāpayām āsa vai || ⊙ || sidaḥ urã̄ Pāṇḍuraṅga niy kin(t)u sadākāla mūrkha dā ja jhāk
	Line 4		buddhi Pāpakarmma măn ha artha nan urã̄ Pāṇḍuraṅga nĭ kā lo vvāra clăṅ adhama kăn pu Pŏ tana rayā nagara campa dadăndadănrāja marai tra | navai
	Line 5		yã̄ Po ku śrī parameśvaravarmmadeva ya tu vuḥ yã̄ pu Po ku parameśvara sakala kā putau jĕ sa Pŏ drī tathāpi la urã̄ Pāṇḍuraṅga nĭ nau Pāpaka

Figure 3.13: Sample ground truth of text line transliteration

ing transliteration. So far, the dataset **Transliteration-Manuscript-Cham** consists of approximately 4507 text lines. One of the most challenging problems when preparing the ground truth for the line of text transliteration task is transcribing some similar characters. In some cases, the transliteration of these characters needs to consider the surrounding context to correctly translate. Moreover, in the Cham writing system, there is no space between words, so the separation between the words is also provided by our expert. Furthermore, there are some words or some transliterations of characters that are not clear, even our experts can not read them, and the study of these words is still in progress. Table 3.1 below lists the characteristics of the datasets created to evaluate the proposed methods.

3.5 Conclusions

In this chapter, we present the kingdom of Champa: its history, its formation, and its development. Its long history and historical events explain the origin and evolution of the Cham language over time.

In this work, two collections are presented namely: Cham inscriptions and Cham manuscripts. The two collections come from different periods in the history of Cham language creating various challenges when processing them. While Cham manuscript structure is quite similar to some other datasets such as Balinese Palm Leaf Manuscripts Kesiman (2018), Cham inscrip-

Dataset Name	Type of document	Period	Evaluation Task	Chapter	Number of sample	Annotation
Denoising-Inscription-Cham	Inscription	6^{th}-15^{th}	Image enhancement	4	190	Text line
Textline-Inscription-Cham	Inscription	6^{th}-15^{th}	Text line segmentation	5	395	Text line
Textline-Manuscript-Cham	Manuscript	18^{th}	Text line segmentation	5	600	Not yet
Transliteration-Manuscript-Cham	Manuscript	18^{th}	Text line transliteration	6	4507	Text line

Table 3.1: Summary of four created datasets

tions are different. The physical support where the texts are written as well as the quality of the documents bring their challenges during the application of the DIA.

To evaluate different methods in DIA, huge efforts from a linguistic point of view have been made such as data collection, data preparation, and ground truth preparation. Thanks to the interaction between our linguistic expert and our computer science and data science team, the ground truths specific to the different tasks in DIA have been constructed.

CHAPTER 4

Image Enhancement

In this chapter, we present the image enhancement problem. The definition of this task, the input, and the desired output are first given. After that, the challenges encountered for Cham inscription have been listed. Based on the initial results from literature works, we present the inspiration of an additional global attention module and the proposed method. The extensive study conducted on the ***Denoising-Inscription-Cham*** *has proved the relevance of the proposed method.*

4.1 Introduction

4.1.1 Definition

Document image enhancement aims to improve the quality of document images by diminishing artifacts such as low contrast and uneven background illumination, bleed-through and shadow effects, damaged characters, and noisy black borders Gatos (2014). Depending on the input document and output requirements, different approaches can be applied. In this work, we formula

document image enhancement problem as: from input documents whose high degradation, the system needs to improve the visual quality of these documents by removing the unwanted parts as well as enhancing the contrast between the background and the characters parts (See Fig. 4.1).

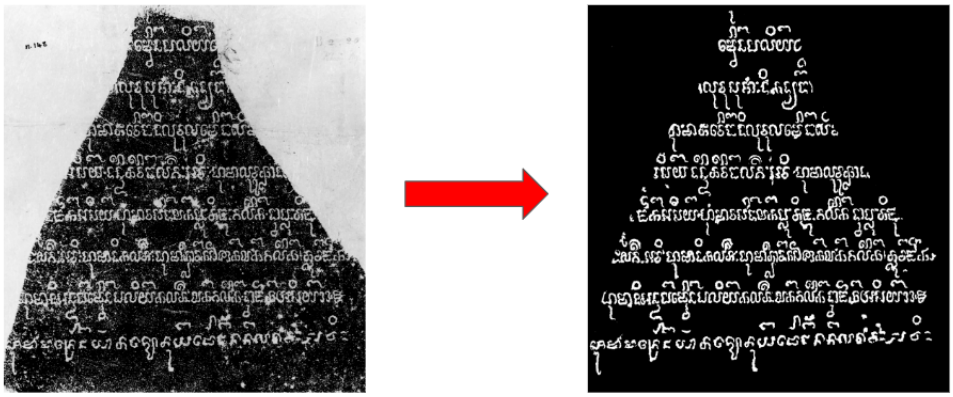


Figure 4.1: Illustration of document image enhancement on the inscription

4.1.2 Challenges

As mentioned above, in this task, we only aim at the processing of the inscription dataset, not the manuscript dataset. Hence, the challenges below mainly come from this dataset. Specifically, several unconventional issues include:

1. **Preservation and Digitization**: Inscriptions engraved between the 6^{th} and 15^{th} centuries have not been well maintained. Due to the aging of these inscriptions and the climatic conditions, the characters were damaged and created bumps. Many unwanted parts or gaps appeared on the stones, which significantly degraded the visual quality of the image. The readability has become a real challenge for archaeologists, historians, as well as for people curious about Cham culture. As the discussion about the digitization process of the inscription in the section 3.2, they also created unwanted parts (which can be considered as the noises), even sometimes losing part of the letters, making it difficult to analyze the Cham text and even build a relevant ground truth.

2. **Variation documents**: Due to the wide variety of documents in the collection from a

chronological range of 6^{th} to the 15^{th} century AD) and the evolution of the writing system (Cham or Sanskrit), the use of each of these two languages leads to a slightly different writing system. For example, in figure 4.2, the "dot" on the left is the noise part while on the right it is the part of the letter. Therefore, the proposed method must have the ability to distinguish not only the noise representations but also the text representations.

3. **Ground truth construction**: Since no dataset and ground truth were available, so a new dataset **Denoising-Inscription-Cham** had to be built from scratch. However, the cleaning process was too tough (due to the high degradation even the experts could not read some parts) so it was a time-consuming preparation. Consequently, the number of available data was very limited, reducing the ability to huge deep learning models.

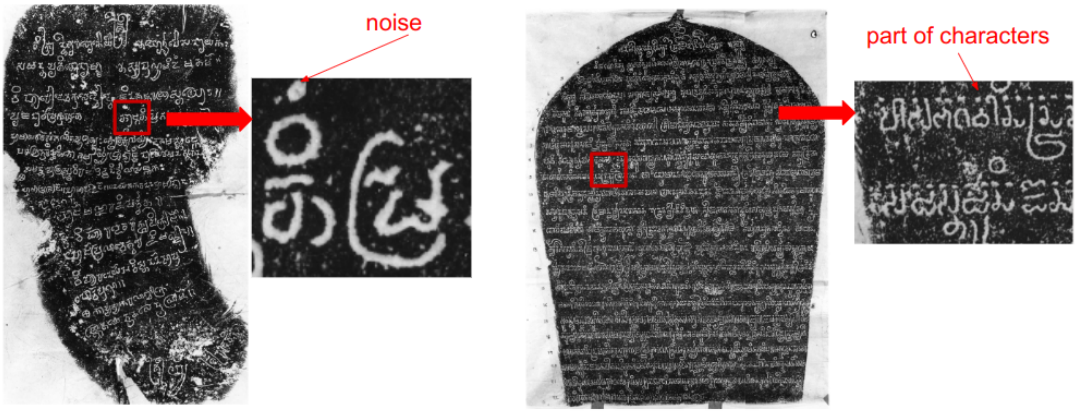


Figure 4.2: Different noises exist in the inscription documents

4.2 Multi-Scale Attention based Encoder Decoder Approach (MSAED)

First, we tried to tackle the problem by using traditional approaches such as Total variation (TV) Rudin et al. (1992) Non-local Means (NLM) Buades et al. (2005), Independent Component Analysis (ICA) Hyvärinen et al. (1998), Block-matching and 3D filtering (BM3D) Dabov et al. (2007), the denoised performance was very limited due to the specific features of the inscriptions mentioned above (See Fig. 4.3). Different types of noise are present in documents. Some are similar to Gaussian noise and can easily be reduced if the correct estimation of sigma noise is

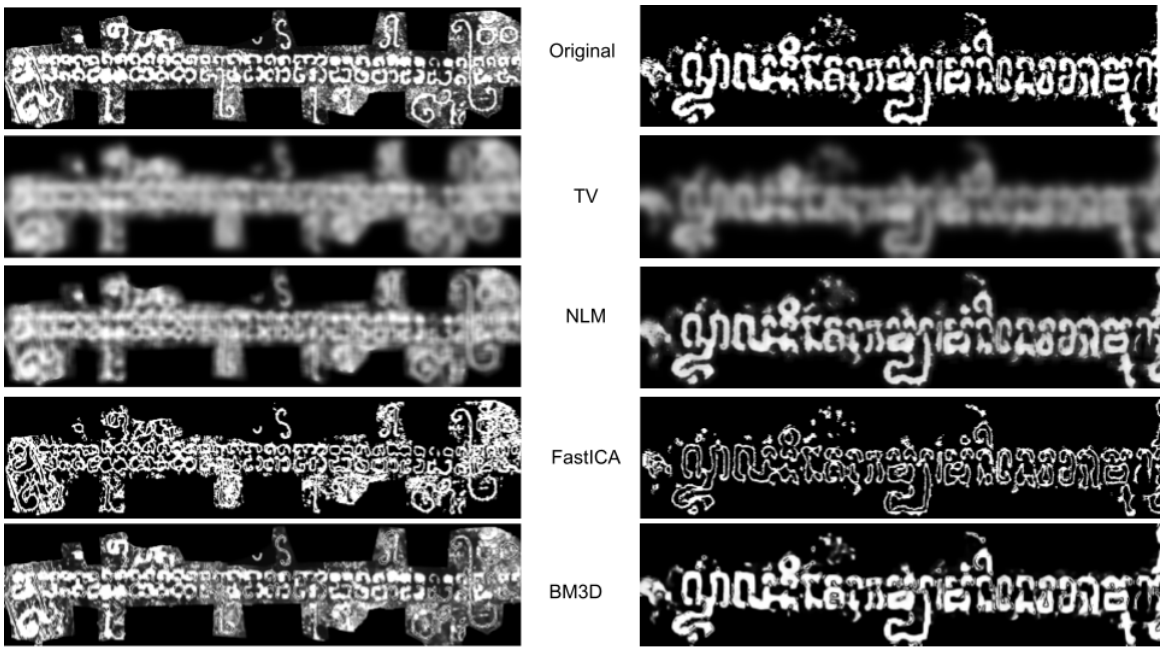


Figure 4.3: Denoising performance with different approaches on the inscription images

provided. But the others are more complicated to eliminate. So, we then choose to focus on the deep learning based approach. Several kinds of models can be used for this problem such as multi Multi Layer Perception (MLP) Burger et al. (2012) or Convolutional Neural Networks (CNNs) Mao et al. (2016), Zhang et al. (2017). Recently, Unet Ronneberger et al. (2015) which is constituted of an encoder-decoder model, obtained good results in the reconstruction image task, so we decided to adapt it as the baseline model. Generally, we noted that the method based on this current approach considers noise and characters equally, this result makes the next steps of the pipeline difficult because some of the letters are missing or unclear with background parts. So we want the model to focus more on the character part, to be separated from the noise parts.

Attention is an interesting module to make the model robust to salient information rather than learning insignificant background parts in the image. Attention has performed well in natural language processing problems such as Luong et al. (2015a); Vaswani et al. (2017) and is gradually being used in computer vision problems Dosovitskiy et al. (2020). Instead of using attention at one unique level scale, different levels of attention have been investigated. At multiple-level scales, different information can be highlighted. At a lower scale, it reflects general information structure in the image while it highlights detailed information at a higher scale Johnson

et al. (2016). Therefore, if we can exploit this important information at different scales, we can provide more detailed information to the model. To efficiently use this information, we propose a global attention fusion mechanism that accumulates the attention from different scales to enhance denoised image quality and further aim at the characters.

4.2.1 Architecture

As shown in Figure 4.4, the proposed model consists of two main components: a baseline encoder-decoder model and the attention module.

Baseline model: The baseline of the proposed model is inspired by Unet Ronneberger et al. (2015) that follows the encoder-decoder architecture. The input image is first processed by a consecutive Convolution-BatchNorm-Relu layers (down-sampling step) to the bottleneck. Then from the output of bottleneck layer, reconstructed image is achieved by doing a sequence of Deconvolution-BatchNorm-Relu layers (up-sampling step). However, the up-sampling step is done from the bottleneck layer where the size of the data is small, much of the information will be lost in the reconstructed images. Instead of down-sampling the input image to the 1x1 feature size, the input image is down-sampled twice before going through a sequence of Resnet block He et al. (2015) in the middle (the blue blocks in Figure 4.4). This type of architecture is similar to the coarse-to-fine generator proposed in Wang et al. (2018).

Attention: To reduce the redundancy of the information when using skip connection Mao et

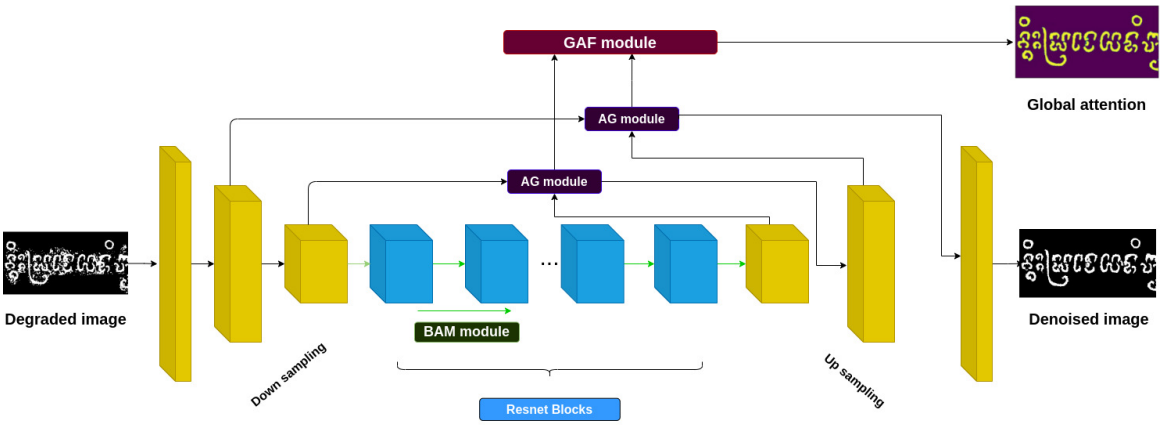


Figure 4.4: Architecture of the proposed model.

al. (2016) between down-sampling steps and up-sampling steps, we adapt the Attention Gated (AG) introduced in Schlemper et al. (2019). This module selects appropriate information instead of keeping all information from the down-sampling step. The integration of the attention gate in skip connection can be explained as below. First, an attention map is generated from F_{D2}

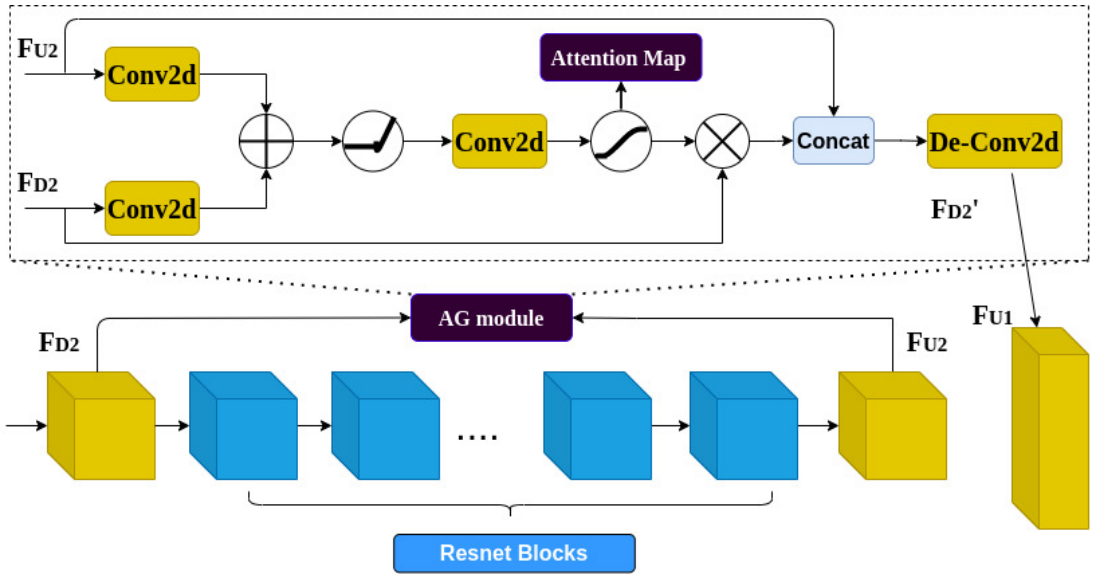


Figure 4.5: Integration of Attention Gated (AG) in Skip Connection Schlemper et al. (2019)

and F_{U2} (last down sampling features (before Resnet block) and first upsampling features (after Resnet block)). This attention map has high values at the positions where similar features are between F_{D2} and F_{U2}. The modified features F'_{D2} are computed by multiplying F_{D2} with the attention map.

$$F'_{D2} = F_{D2} * \sigma(W_n * (\omega((W_d * F_{D2} + W_u * F_{U2}))) \tag{4.1}$$

where F_{D2}, F_{U2} are the features at the downsampling step and upsampling step respectively, W_d, W_u and W_n are the parameters of the convolution layers, ω is Relu activation function, σ is the Sigmoid activation function. After that, we concatenate F'_{D2} and F_{U2} features as common skip connection then up-sample it to F_{U1}.

$$F_{U1} = W_t * (concat([F_{U2}, F_{D2'}])) \tag{4.2}$$

where W_t are the parameters of the de-convolution layer. This process is also repeated at the next upper scale.

The details of the module are represented on Figure 4.5.

For the sequence Resnet block in the middle of our architecture, we modify the normal connection between Resnet blocks by adding the Bottleneck Attention Module (BAM) Park et al. (2018) after every Resnet block to improve the separation of the features at low-level such as: gaps or strokes on the surface and gradually focus on the exact target at a high level of semantic (characters) by integrating attention from both channel and spatial information. Attention Channel reflects the relationships between channels, where features should be highlighted. Instead of computing attention cross channel-like, spatial attention reflects the important information at spatial locations. Let's assumption the intermediate feature is $F \in R^{C\times H\times W}$ then Channel attention is computed as :

Algorithm 1 Channel attention Park et al. (2018)

Require: $F \in R^{C\times H\times W}$
$F_c = AvgPool(F)$ {Global average pooling over the width and height channel}
$M_c(F) = BN(MLP(F_c))$ {MLP is multi-layer perception with one hidden layer, BN is Batch Norm layer}

Spatial attention is computed as :

Algorithm 2 Spatial attention Park et al. (2018)

Require: $F \in R^{C\times H\times W}$
$F_1 = f_o^{1\times 1}(F)$ {$f_o^{1\times 1}$ denotes 1×1 convolution}
$F_2 = f_2^{3\times 3}(f_1^{3x3}(F_1))$ {$f_2^{3\times 3}$, $f_1^{3\times 3}$ denotes 3×3 dilated convolutions}
$M_s(F) = BN(f_3^{1\times 1}(F_2))$ {$f_3^{1\times 1}$ denotes a convolution operation, BN is Batch Norm layer}

An illustration of the computation of two attention is shown in figure 4.6. For our model, BAM module is integrated after every Resnet block. Every Resnet block output feature is gradually refined by the BAM module. After the sequence Resnet block + BAM (see Fig. 4.7), the model aims to highlight salient features of character regions while deducting the features of non-relevant regions. The refined features by BAM can be described as:

$$F' = F + F * \sigma(M_c(F) + M_s(F)) \tag{4.3}$$

where F is the output feature of the previous Resnet block. M_c, M_s are the channel attention and spatial attention (computed in the algorithm 1, algorithm 2), respectively.

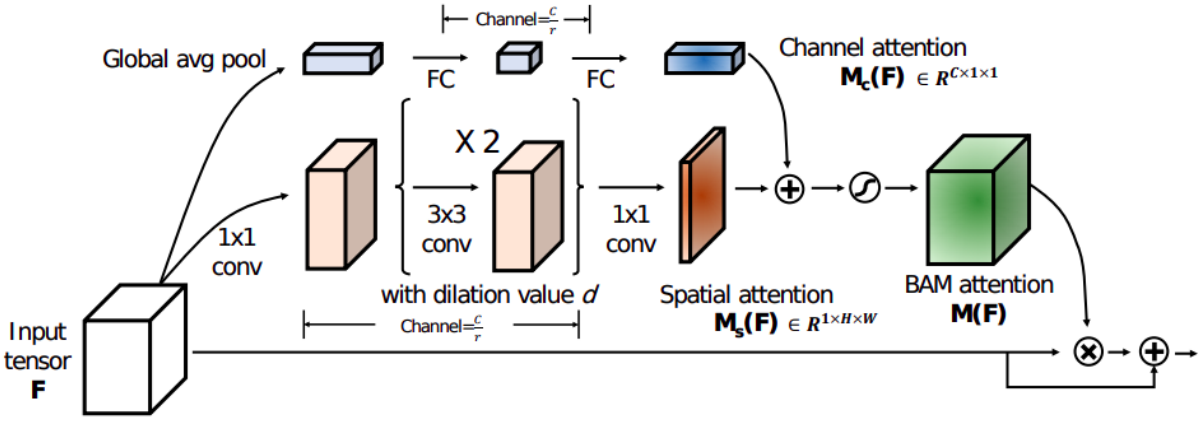


Figure 4.6: Computation of channel and spatial attention Park et al. (2018)

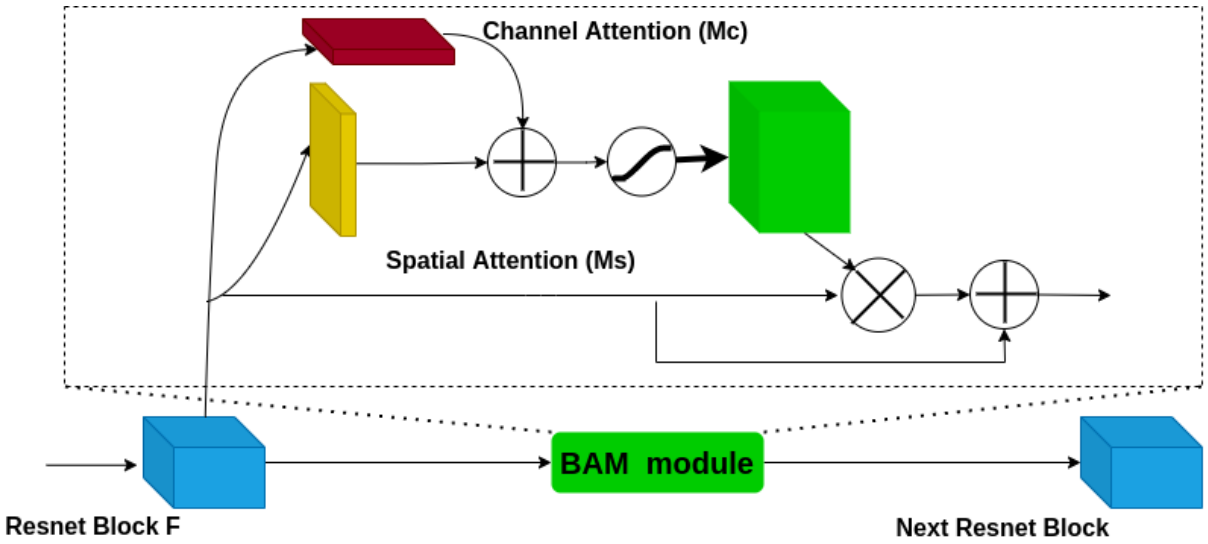


Figure 4.7: Refined features by BAM Park et al. (2018) on our model

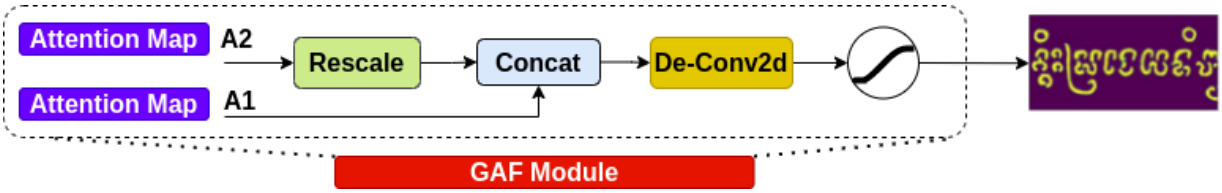


Figure 4.8: Global Attention Fusion Module

Global Attention Fusion: Due to the difficulty when reading Cham inscriptions, it is more important to keep and enhance the quality of the characters than remove only the noise. Therefore, we propose a global attention fusion module by integrating attention at multiple scales to help the model focus more on the pixels of the characters. This module works as follows. First, we concatenate the attention map from different scales of the attention gate module.

$$C(A) = concat(A_1, resize(A_2)) \tag{4.4}$$

where A_i be the attention map generated from the attention gate at different scales, i is the scale of the input image. The concatenated attention is then followed by a de-convolution layer to generate the global attention map which has the same shape of the input image. Then Sigmoid activation function is used to normalize the coefficients of the global attention to the range (0,1).

where W_A is the parameters of de-convolution layer, σ is the Sigmoid activation function. (See Fig. 4.8). These coefficients of the global attention map (C_s) represent the confidence of the generated pixels for the character regions in the image. As we can see in Figure 4.9, at the beginning, only simple patterns, representing noise and characters, whose model is easy to recognize will be correctly reconstructed with a high confidence score. After some epochs, this score gradually increases to reveal difficult patterns of the characters that have lower scores at the first epochs.

4.2.2 Objective Function

The training objective function combines two main functions. The first function, weighted L1 Loss (weight reconstruction loss), helps to reconstruct the denoised image of important regions

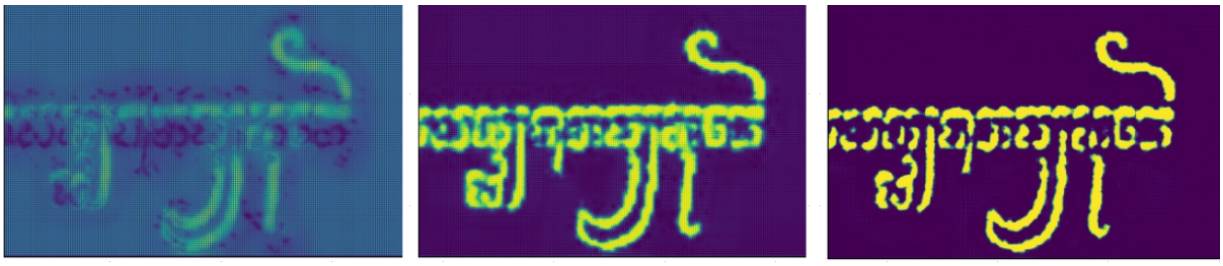


Figure 4.9: Evolution of confidence score. Bright pixels have a high score, dark ones have a low score.

closest to the ground truth images.

$$L_1^w = ||C_s * \hat{y} - y||_1 \tag{4.5}$$

where C_s is the confidence score output from GAF module. The second one is the perceptual loss L_p Johnson et al. (2016) that measures high-level perceptual and semantic differences between the reconstructed image and ground truth image. Instead of comparing them at the pixel level, this loss function will encourage the model to construct image close to the ground truth image at the high level such as: structure, and content. Perceptual loss is computed through the perceptual features. These features can be found through the intermediate features extracted from different layers of a pre-trained CNNs model such as VGG Simonyan and Zisserman (2014), Inception Szegedy et al. (2015) or Resnet He et al. (2015). Following the proposal of the Johnson et al. (2016), we adapt VGG-16 as a feature extractor.

$$L_p = \sum_{i=1}^{n} \frac{1}{C_i * H_i * W_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \sum_{c=1}^{C_i} ||F^i_{h,w,c}(\hat{y}) - F^i_{h,w,c}(y)||_1 \tag{4.6}$$

where $F^i_{h,w,c}$ is output feature from list n features named as 'relu1-2', 'relu2-2', 'relu3-3', 'relu4-3' of VGG-16 network.

We found that if we used C_s in the equation 4.5 without applying any constraint, the coefficients of C_s have high values in regions almost empty because, for those regions, the model is easy to reconstruct. After that, the optimization for these regions will not change. (See Fig. 4.10). To

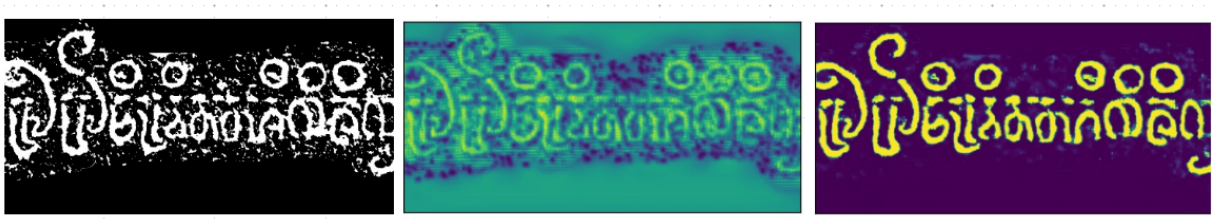


Figure 4.10: Attention map with and without constraint on the C_s. We can see that in the middle of the figure when the model is trained without constraint, character regions, and empty regions have high coefficients. In the right image, when using constraints, the model aims to reveal only character regions, the other regions have a low score.

avoid that we simply use the mean of C_s as an additional constraint that helps C_s to consider only the character regions instead of the empty regions.

Overall the objective function can be described as:

$$L = \lambda_{l_1} * L_1^w + \lambda_{l_{cs}} * mean(C_s) + \lambda_{l_p} * L_p \tag{4.7}$$

where λ_{l_1}, $\lambda_{l_{cs}}$ and λ_{l_p} are the weights of the weight reconstruction loss (L_1^w), the mean value of final attention map (C_s) and the perceptual loss (L_p) in the total loss, respectively. Each weight indicates the contribution of each component in the final loss function. The higher the weight is, the greater the contribution of the components is. The model needs to balance between pixel level and high feature level based on the different weights of each loss.

4.3 Training Setting

4.3.1 Data preparation

In this task, we evaluate the proposed model on the **Denoising-Inscription-Cham** presented in 3.2. Due to the different sizes of each text line image, we normalized the size of all training images. Instead of resizing the whole text line image which leads to distortion, we simply cropped the original image into sub-images which have a size of 256x512 pixels (See Fig. 4.11). Pair sample of training data can be found in figure 4.12.

Figure 4.11: Line cropped in a sub-image whose size is 256x512

Figure 4.12: Pair samples of training data. Top : the degraded image ; Bottom : the corresponding clean image

Overall, 140 text line images were split into approximately 2500 images as training data while 14, 50 other text line images were used as validation and testing data.

4.3.2 Parameters Settings

All models are trained from scratch. Due to the limited amount of data available, the augmentation strategies were also studied. The following data augmentation methods have been used: random rotation (sample images will be randomly rotated from -90 to 90 degrees), random erasing (randomly replace a region in the image by new intensity values), similar mosaic augmentation Bochkovskiy et al. (2020) (combining images by cutting parts from some regions and pasting them on the augmented image). Each method is used with a probability of 30%, only random erasing has a probability of 10%.

We have some experiments to see the change in results when setup with different weights of each loss component. When we set more value on the reconstruction loss λ_{l_1} that forces the model to generate an image closest to the ground truth image but the results do not change even worse than the lower value. This can explained by the ground truth preparation, the pixel of noisy image and ground truth image sometimes are not homogeneous, so the more value on this weight, the more over-fitting on the trained model. So the weight of the reconstruction loss λ_{l_1} and λ_{l_p} are set too close. For the weight $\lambda_{l_{cs}}$, if we set a huge value, due to the optimization process, it will force all pixels to be equal to zero. On the contrary, the small value will force all pixels to be equal to one, we will lose the proposal meaning of this module. The weight λ_{l_1}, $\lambda_{l_{cs}}$ and λ_{l_p} were then determined 1, 0.1, 2, respectively.

For the general training, we used the Adam optimizer with an initial learning rate 0.0002. All models in our experiments were trained from scratch with the same number of training data. The experiments were done on a processor 2080Ti GPU with 12GB of memory.

4.4 Metrics and Evaluation

4.4.1 Metrics

In order to evaluate the denoising performance, we used two metrics: Structural Similarity Index Measure (SSIM) Wang et al. (2004) and Peak Signal-to-Noise Ratio (PSNR).

PSNR is a quality metric that measures how the denoised image is close to the ground truth image. PSNR can be considered as an approximation to human perception to estimate the quality of the reconstruction. PSNR is computed through Mean Square Error (MSE). MSE is defined as below:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} [y(i,j) - \hat{y}(i,j)] \tag{4.8}$$

where $y(i,j)$, $\hat{y}(i,j)$ are the ground truth image, the reconstructed image, respectively. Then PSNR is computed as:

$$PSNR = 10log_{10}(\frac{MAX_y^2}{MSE}) \tag{4.9}$$

where MAX_y is the maximum pixel value of the image.

SSIM metric measures the difference between the reconstructed image $\hat{y}$ and the reference image image y using a variety of well-known properties of the human visual system. SSIM is weighted of those three comparative measures: luminance (l), contrast (c), and structure (s).

$$l(y,\hat{y}) = \frac{2\mu_y\mu_{\hat{y}} + c_1}{\mu_y^2 + \mu_{\hat{y}^2} + c1} \tag{4.10}$$

where μ_y, $\mu_{\hat{y}}$ is the mean of pixels of y, $\hat{y}$.

$$c(y,\hat{y}) = \frac{2\sigma_y\sigma_{\hat{y}} + c_2}{\sigma_y^2 + \sigma_{\hat{y}}^2 + c_2} \tag{4.11}$$

where σ_y, $\sigma_{\hat{y}}$ is the variance of pixels of y, $\hat{y}$

$$s(y,\hat{y}) = \frac{\sigma_{y\hat{y}} + c_3}{\sigma_y\sigma_{\hat{y}} + c_3} \tag{4.12}$$

where $\sigma_{y\hat{y}}$ is the covariance of y and $\hat{y}$.

The SSIM is then computed as follows :

$$SSIM(y, \hat{y}) = l(y, \hat{y})^{\alpha} \times c(y, \hat{y})^{\beta} \times s(y, \hat{y})^{\gamma} \tag{4.13}$$

where α, β, γ is set to 1.

The highest PSNR and SSIM values indicate the better results.

4.4.2 Experiments

In this section, we conducted experiments to evaluate the different aspects of the proposed method. Firstly, we carried out an ablation study to analyze the contribution of the different modules of our model. In the two next experiments, we compare the proposed method with other approaches in the literature. To evaluate the generalization ability of the proposed method, an experiment on the whole dataset is presented. Finally, the results with a weakly trained process are given.

Ablation study

In the experiments, we want to determine the contribution of each module to the whole model. Four different models have been trained. The first model is a simple baseline model (Encoder-Decoder module). After that, the model was trained by adding different modules to the baseline model: Attention Gated (AG), Bottleneck Attention Module (BAM).Finally, the proposed model is evaluated. Table 4.1 presents the results of the ablation study and shows the effects of

Table 4.1: Ablation study on the sub-components of the proposed model

Condition	PSNR	SSIM
Baseline model	15.45 ± 2.65	0.898 ± 0.05
Baseline model + AG	15.57 ± 2.62	0.900 ± 0.049
Baseline model + BAM	15.86 ± 2.76	0.904 ± 0.048
Proposed model (MSAED)	**15.98 ± 2.82**	**0.905 ± 0.048**

each component. On the SSIM metric, the results are similar because, on text line images, the structure or contrast of the image is quite simple, therefore all methods can simply preserve the original structure. On the PSNR metric, we can see that the use of the AG module and BAM

module have slightly improved results. The proposed method can achieve better results than each module combined separately and shows an improvement in comparison to the baseline model.

Comparison with other approaches of the literature

To evaluate the relevance of the proposed approach, we have compared, in this experiment, the proposed method to several approaches in the literature. The summary results can be seen in the Table 4.2. To be easy to observe, we split these methods into two main groups. In the first group,

Table 4.2: Quantitative results on our dataset

Method	Avg-PSNR	Avg-SSIM
Original	11.93	0.651
TV Rudin et al. (1992)	12.68	0.567
NLM Buades et al. (2005)	12.01	0.489
BM3D Dabov et al. (2007)	11.64	0.623
Ostu Otsu (1979)	10.96	0.713
Sauvola Sauvola and Pietikäinen (2000)	10.44	0.689
Niblack Niblack (1985)	10.59	0.694
NMF Févotte and Idier (2011)	12.63	0.749
FastICA Hyvärinen et al. (1998)	12.69	0.754
Unet Ronneberger et al. (2015)	15.54	0.900
Pix2pix Isola et al. (2017)	14.05	0.875
Pix2pixHD Wang et al. (2018)	13.26	0.855
Proposed method (MSAED)	**15.98**	**0.905**

the input image is directly processed without any training step. The second group consists of methods with a training step integrating knowledge from both the original (degraded image) and cleaned image. The first group gathers traditional denoising methods (TV Rudin et al. (1992), NLM Buades et al. (2005), BM3D Dabov et al. (2007)) and binarization methods (Otsu Otsu (1979), Sauvola Sauvola and Pietikäinen (2000), Niblack Niblack (1985)).

For the traditional methods, on average, PSNR is improved but the value of the SSIM metric is lower than the original images. This can be explained by the fact that these methods are efficient in regions where the size of the noise is relatively small, but it leads to the output image blurrier. The results with binarization methods show that these approaches are not adapted since they do not reduce the noise (lower PSNR than the original image) even if they enhance the global visual

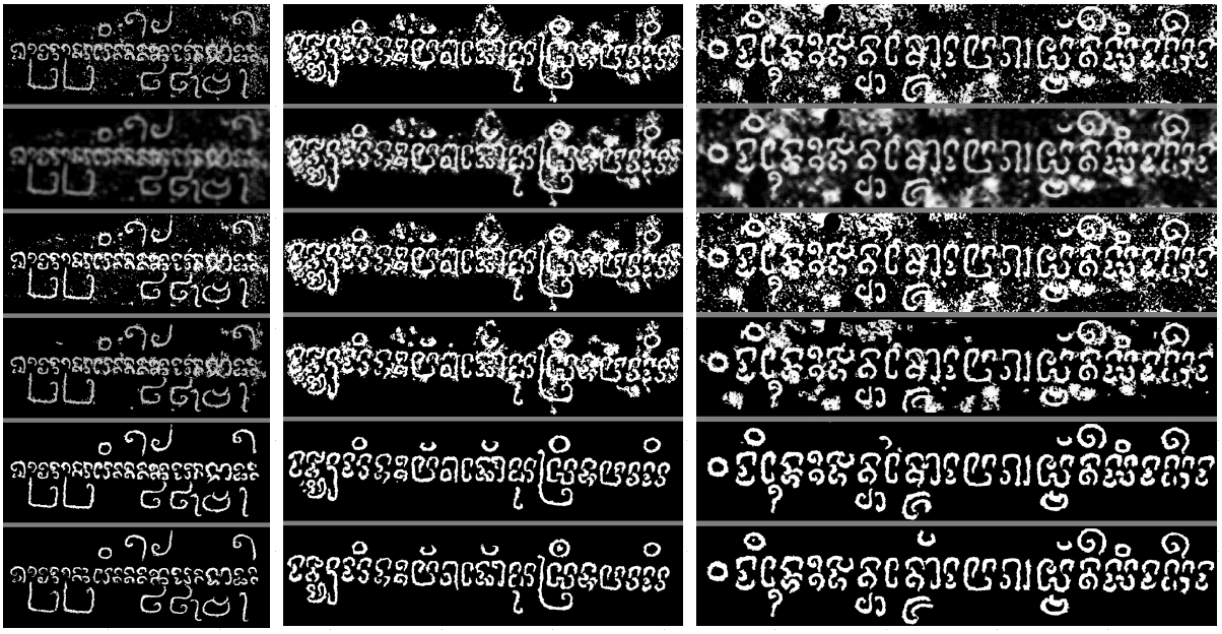


Figure 4.13: Qualitative results on **Denoising-Inscription-Cham** with **low** degradation. From top to bottom: Original image, NLM, Otsu, FastICA, Proposed method, Ground truth.

quality of the image providing a better SSIM than the original image. The qualitative results on the images with the different methods on low and high degradations are shown respectively in Figure 4.13 and Figure 4.14.

In the second group, both NMF Févotte and Idier (2011) and FastICA Hyvärinen et al. (1998) methods improve both the PSNR and SSIM metrics but we observed that the denoising ability is very limited when the patterns of noise are similar to some parts of the characters affecting the readability. The last methods of this second group consist of deep learning-based approaches. All methods have significantly boosted both the quantitative (PSNR and SSIM metrics) and qualitative results. The proposed method achieves better results on both metrics. In addition to improving the denoising results compared to the other methods, our approach gave better results because it generates pixels with a high confidence value in the foreground and thus provides a better visual quality.

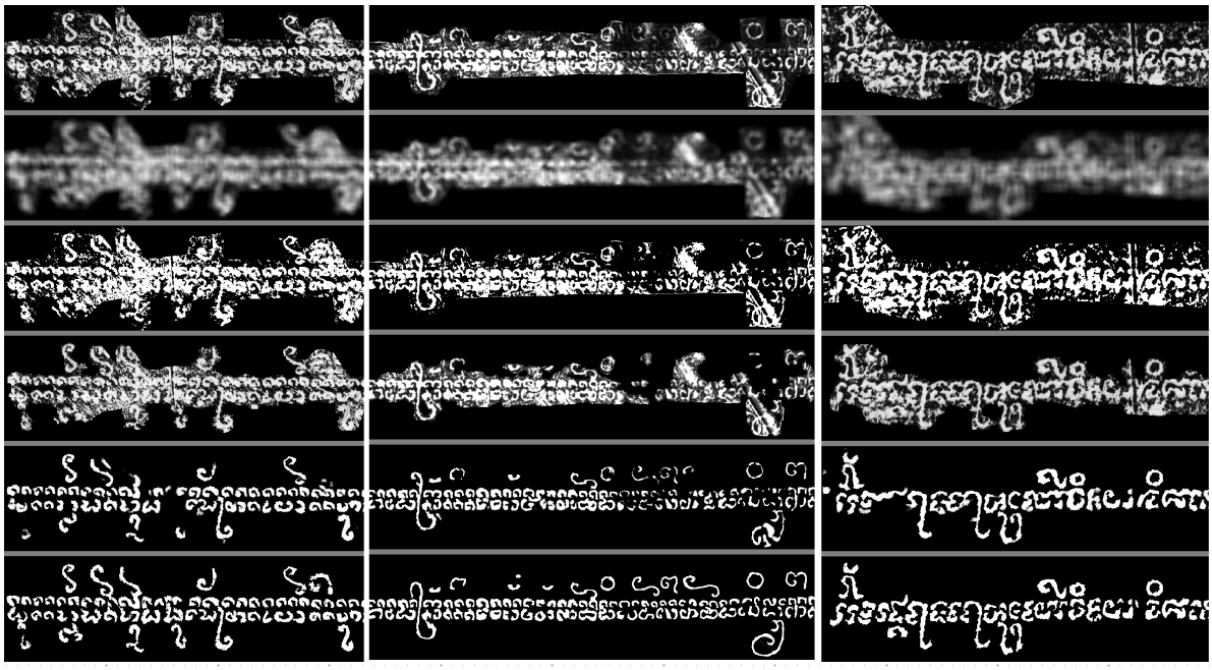


Figure 4.14: Qualitative results on **Denoising-Inscription-Cham** with **high** degradations. From top to bottom: Original image, NLM, Otsu, FastICA, Proposed method, Ground truth.

Qualitative results

To further evaluation the quality of the image, the readability is also an important parameter for the next step: the recognition process. To evaluate the relevance of the approach we asked an expert in Cham language to estimate the performance of our method. We asked her to analyze qualitatively two different criteria: performance on noise removal and character readability. For each criterion, we defined four level assessments: very bad, bad, normal, and good, corresponding respectively to the 1, 2, 3, 4 scores. The qualitative evaluation is obtained by computing, for each experiment , the average score for each criterion.

However, instead of evaluating the results on the whole testing set, it was split into three subsets depending on the historical period: $7^{th} - 9^{th}$ century (5 images), $10^{th} - 12^{th}$ century (19 images) and $13^{th} - 15^{th}$ century (26 images). The quality of the inscriptions depends on their age since the damages are more important on the older stones. The separation of images into chronological categories has been motivated by the evolution of writing, starting from often irregular and less codified scripts, passing through the blossoming and mastery of characters, and ending with less refined scripts and less distinguished characters, creating more confusion for deciphering. So,

Table 4.3: Average score of the qualitative evaluation on our testing dataset split in 3 sets depending of the historical period

Aspects	Condition	$7^{th} - 9^{th}$	$10^{th} - 12^{th}$	$13^{th} - 15^{th}$
Noise Removal	Baseline	3 ± 0	2.45 ± 0.6	1.53 ± 0.58
	Baseline + BAM	3 ± 0	2.8 ± 0.52	2.31 ± 0.68
	Baseline + AG	3 ± 0	2.6 ± 0.5	1.85 ± 0.61
	Proposed method (MSAED)	**3.4** ±0.55	**2.84** ± 0.66	**2.65** ± 0.77
Character Readability	Baseline	2.2 ± 0.45	1.94 ± 0.41	1.34 ± 0.41
	Baseline + BAM	3 ± 0.7	2.05 ± 0.4	1.88 ± 0.41
	Baseline + AG	3.2 ± 0.45	2.15 ± 0.5	1.96 ± 0.41
	Proposed method (MSAED)	**3.4** ± 0.55	**2.52** ± 0.77	**2.34** ±0.41

we created 3 categories by chronological separation.

Table 4.3 presents the qualitative results. For the noise removal criterion, we observe that the integration of the attention module (AG or BAM) into the baseline model improves significantly the results compared to the simple baseline model. For the character readability criterion, the proposed method outperforms the other approaches. This qualitative evaluation shows the effect of the GAF module by encouraging the model to generate character pixels with higher confidence values. The improvement of contrast between the foreground and background part as well as the quality of the characters, allows to increase the readability of the Cham inscription.

Full documents evaluation

In order to evaluate the generalization capacity of the model, we evaluated the performance of the proposed model on the complete inscriptions instead of only "text-line" image as in the previous experiments.

We noted that the proposed model can adapt itself by well reconstructed the main parts of the documents. Document enhancement capability is quite good but there are still some issues in some cases. First, there still exists some long and big straight strokes inside or at the border like in figure 4.15 that model can not remove properly. Second, some parts of the letters can not be well reconstructed and sometimes erroneous deletions appear in case of high degradation as shown in figure 4.16. Two kinds of issues come from the limitation of training data. For the first issue, training data only contains the text line which does not include the border pattern, hence

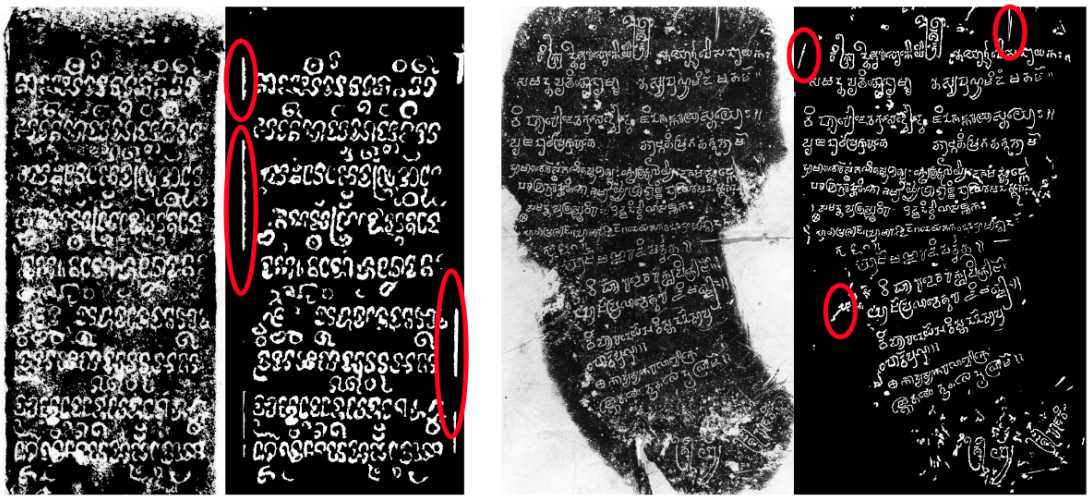


Figure 4.15: Evaluation of image enhancement on the complete inscription images. Red circles indicate the incomplete removal of unwanted parts.

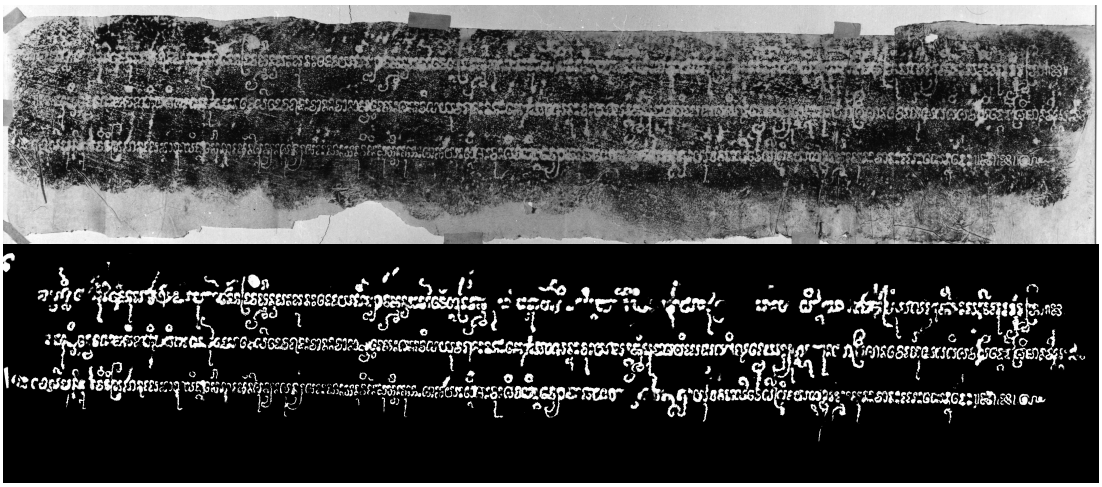


Figure 4.16: High degradation inscription with incorrect removals.

model can not handle it well. The second issue can be explained by the fact that in case of high degradation, even the expert can not read the extracted parts of letters.

Weakly training evaluation

In this section, we try to use a weak training scheme to train the model to reduce the cost of preparing the ground truth for a huge amount of data. The weak training scheme is first to train the model on a small dataset with an annotated ground truth then use the trained model to predict for new sample on unlabelled data. The predicted data then combined with labelled data continue to train the model. Weakly training is often used in the semantic segmentation Ahn and Kwak (2018), Ahn et al. (2019) task. The difficulty in applying this approach is to determine the policy for combining unlabelled and labeled data, otherwise, the performance of the model can be degraded since bad samples could lead to a worse model. In our model, the confidence score C_s indicates how confident the model is on the predicted pixels. By using this inspiration, we use this score as a measure to determine whether new samples are selected to train the model. A new sample is added when the average score of white pixels in the sample is over a given threshold λ_c. We found that the determination of λ_c is very important for the performance of the model. If the threshold is too low, the new sample will not be good enough, whereas, with too high value, few samples will be added. In the beginning, we set λ_c to 0.9 then until 0.95 too many new samples were added. We observed the performance of the model was significantly degraded after a few training epochs. So, we decide to increase more the threshold and limit the number of new samples. We found that a threshold of 0.97 achieved acceptable results. The details of weakly training can be found in the following algorithm 3. Weakly training process started from the best checkpoint of supervised training. This process is continuously trained in 100 epochs with a learning rate initialized $1e-5$ and decreased. After each 5 epochs, the set of unlabelled data is used to get the update on the training data. The set of unlabelled data is from a set of 78 inscription images (it is worth that one complete inscription can generate many sample training from 60 to 100 sample training). It is worth noting that instead of using all images in one time when computing, 20 random images are randomly selected. This one can prevent additional too much new data which is risky to decrease the performance of the model.

Algorithm 3 Weakly training process

Require: Trained model from labelled data P_w (w represents all parameters of the model) and C_s confidence score, labelled data L=(x_i, y_i), unlabelled data (new samples) ($\tilde{x}$).
repeat
$\tilde{y_i}, \tilde{C_s} \leftarrow P_w(\tilde{x_i})$ {Use trained model to predict new samples}
if $AVERAGE(\tilde{C_s} > 0.5) > 0.97$ **then**
Add pair ($\tilde{x_i}, \tilde{y_i}$) to training data
end if
$P_w \leftarrow P_w(\tilde{L})$ {Fine tuning of the model with new training data (including both new samples and current samples) $\tilde{L}$}
until End of the training

Quantitative results In the first evaluation, we compare the performance of the model without and with additional weak training. The results are given in table 4.4. However, the reconstruction of some letters is not complete in some cases. So the PSNR is slightly improved, but the SSIM is slightly lower. This can be explained by the quality of the data. Even if some samples get high confidence, some regions of the sample are not well reconstructed. Thus, the use of these samples for the training confuses the model, leading to reconstruction errors. The second reason for this problem may come from the features of the characters engraved on the inscriptions of the collection. The new samples can be quite different than the labeled data used for the training. Thus, the model can not take advantage of the new knowledge provided by the new training data and the performance can not be improved too much.

Table 4.4: Quantitative results without and with weakly training

Method	Avg-PSNR	Avg-SSIM
Base model(**Proposed method**)	15.98	0.905
Continuous weakly training	16.00	0.899

Qualitative results In the second evaluation, we compare the performance on the whole inscription. The main comparison can be seen in figure 4.17 and figure 4.18. The weakly training improves the results in comparison with the base model by removing more noise in the image and also keeping some diacritics and some parts of the characters (See Fig. 4.17). We also noted that, when the reconstruction image from the base model is good, the results with weakly training will be enhanced, if the results are not good from the base model, the weakly training

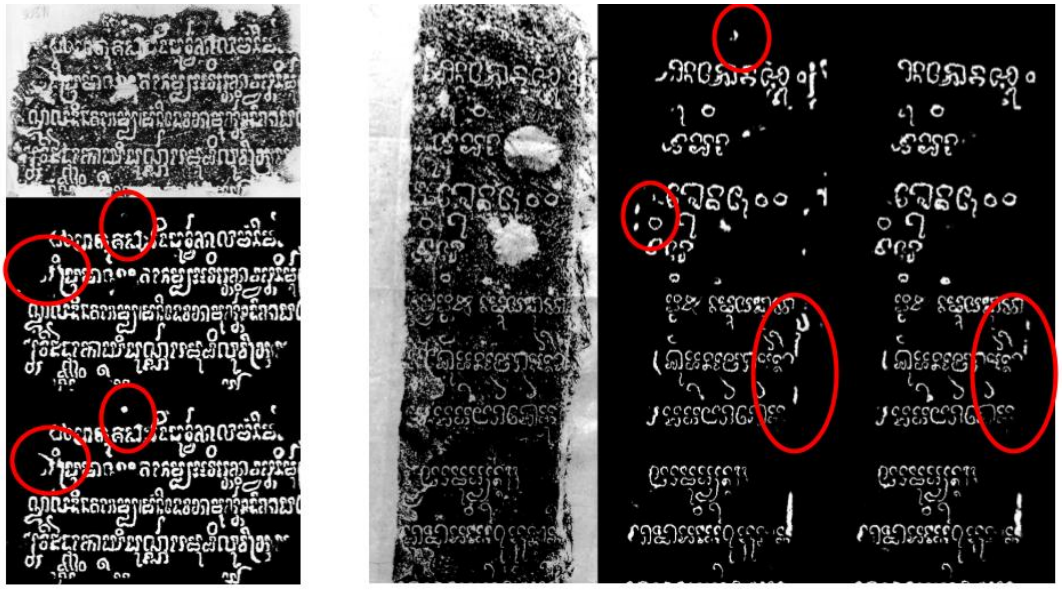


Figure 4.17: Good qualitative results when using weak training. From top to bottom and from left to right: baseline model and weakly denoising results

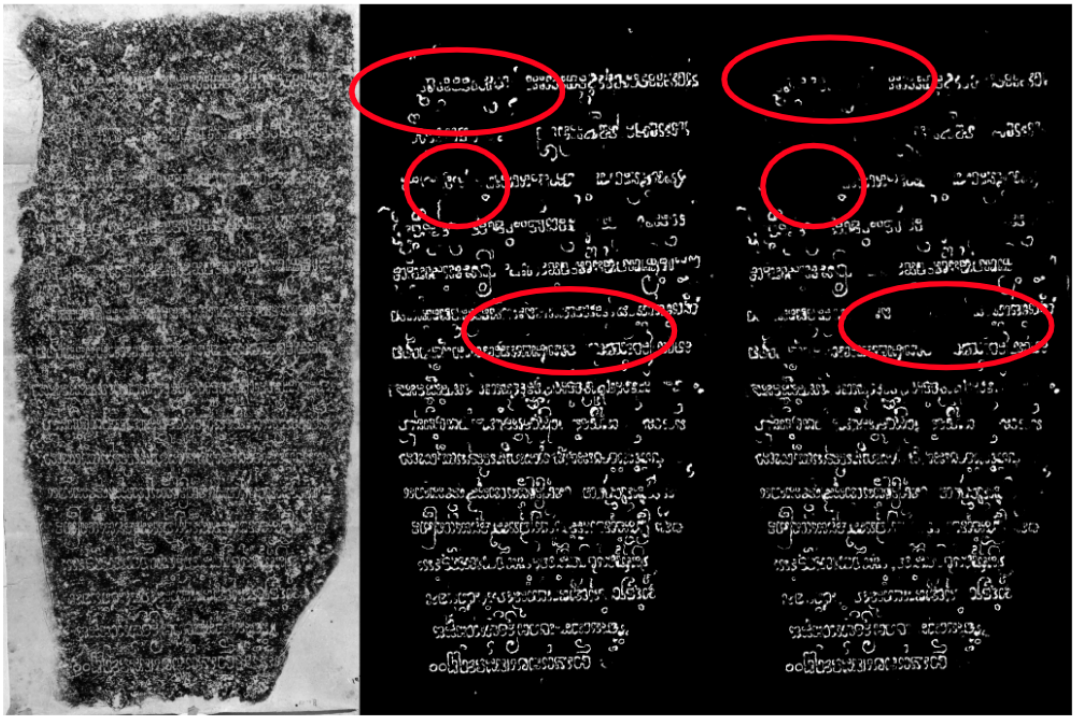


Figure 4.18: Worse qualitative when using weakly training. From the left to right: baseline model and weakly denoising results

leads the model to worse in that case (See Fig.4.18). Therefore the strategy of selecting new samples and the results from the baseline model for additional training is very important. More investigations on the method used to evaluate the quality of the new samples is necessary to process the inscription collection.

4.5 Conclusions

In this chapter, we presented the definition and the challenges of the document image enhancement problem of Cham inscription documents. We tested some approaches in the literature and analyzed the performance of each approach. From these observations, deep learning based strategy has been chosen.

To enhance the quality of output results, an attention module has been integrated to help the model focus on the most important parts of the documents. An ablation study shows the added value of each module to the whole model. The evaluation of the proposed method both qualitatively and quantitatively with other approaches of the literature has proven its ability to remove unwanted parts as well as improve visual quality by reconstructing the characters.

Moreover, the limitation of the proposed method comes mainly from the training data and especially from the characteristics of the samples of the collection and the amount of data available. To solve these problems, two strategies can be studied: preparing more data and using different training strategies with less ground truth training data. The former needs more human resources to annotate manually the data, a task that is costly and time-consuming. Recently, unsupervised approaches have become more interesting, especially for the segmentation problem Kim et al. (2020), Kanezaki (2018). The strategy of using unlabeled data combined with labeled data has also been studied to improve the model. However, further study should be conducted to determine the best criterion for selecting unlabeled data used to refine and improve the model trained with labeled data.

CHAPTER 5

Text line segmentation

In this chapter, we present the text line segmentation problem on the Cham documents. Two kinds of documents with different challenges are studied. Due to the specific characteristics of the writing scripts in Cham documents, we define the problem by considering in detail the specificities of the input and output images. Considering all these special conditions, the general approach that is applied to different datasets is given, and then an adaptation for each dataset is detailed.

5.1 Introduction

5.1.1 Definition

Text line segmentation is a labeling process that consists in assigning the same label to spatially aligned units (such as pixels, connected components, or characteristic points) Likforman-Sulem et al. (2007). As the definition of the task, given a document image that may contain many lines, the text line segmentation methods need to determine the regions of all the lines appearing in

the documents. To be clear about text line region, some components need to be determined (See Fig. 5.1) :

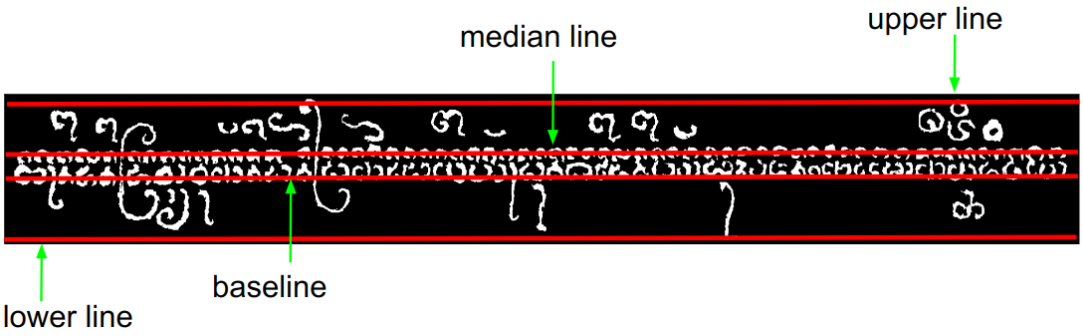


Figure 5.1: Representation of text lines

- Baseline: fictitious line which follows and joins the lower part of the character bodies in a text Likforman-Sulem et al. (2007).
- Median line: fictitious line that follows and joins the upper part of the character bodies in a text line.
- Upper line: fictitious line which joins the top of the *ascenders*.
- Lower line: fictitious line which joins the bottom of the *descenders*.

In typography and handwriting, a *descender* is the portion of a letter that extends below the baseline of a font. Respectively, an *ascender* is the portion of a letter that extends above the median line of a font. A lot of algorithms have been proposed for text line segmentation. Some of them only aim to detect the baseline such as Renton et al. (2017), Feldbach and Tonnies (2001). However, if we only consider baseline segmentation, the next step of recognition can not completely performed due to the absence of the other characteristic components of text lines such as ascenders, descenders, and diacritics. (See Fig. 5.2). Therefore, we consider here that the text line segmentation algorithm must not only correctly segment the baseline but also all the diacritics, ascenders, and descenders belonging to this text line.

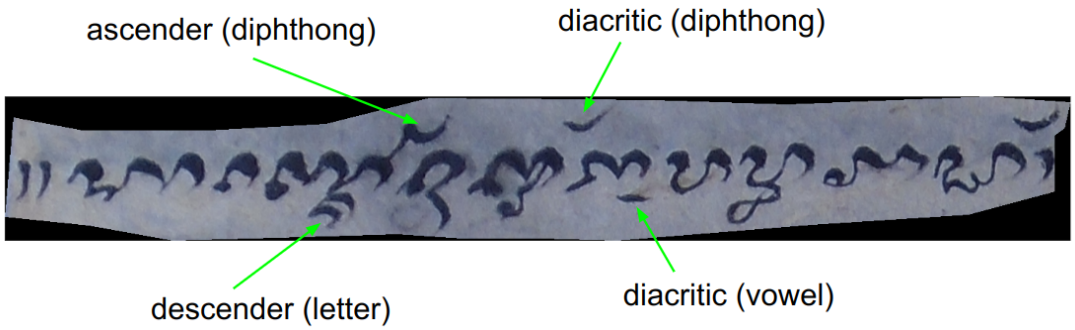


Figure 5.2: Important components (vowel, diphthong, and affixes letter) for the recognition task

5.1.2 Challenges

As discussed in the introduction of this chapter, there are two types of document images: Cham inscriptions and Cham manuscripts whose writing is based on Cham scripts. Cham scripts include many ascenders, descenders, and diacritics which are important, as mentioned above, for proper character recognition. These parts of the characters must be considered when applying the text line segmentation task. While the Cham manuscript documents are quite close to the datasets of the literature dedicated to the text line segmentation problem Kesiman (2018), Valy (2020) (the main difference comes from the sophisticated position of vowels and diphthongs), the Cham inscription documents are totally different, bringing some challenges for this task as described below :

- **Damaged documents**: Due to the high degradation of the documents, the direct use of typical methods of text line segmentation is often ineffective. Hence, the pre-processing steps to clean these documents (image binarization, image enhancement, image denoising) have to be applied. We have observed that the simple process on the degraded inscriptions can be incorrect due to the noise. However, a similar process applied to the enhanced images allows the generation of good candidate lines and thus obtains better segmentation. (See Fig. 5.3).

- **Complexity of Cham scripts**: The characteristics of these letters also present challenges that must be considered. As the inscription documents come from a long period, the sophisticated Cham writing system has evolved. It contains both small, oversized, or mixed characters combined with decoration/notation symbols of the documents leading

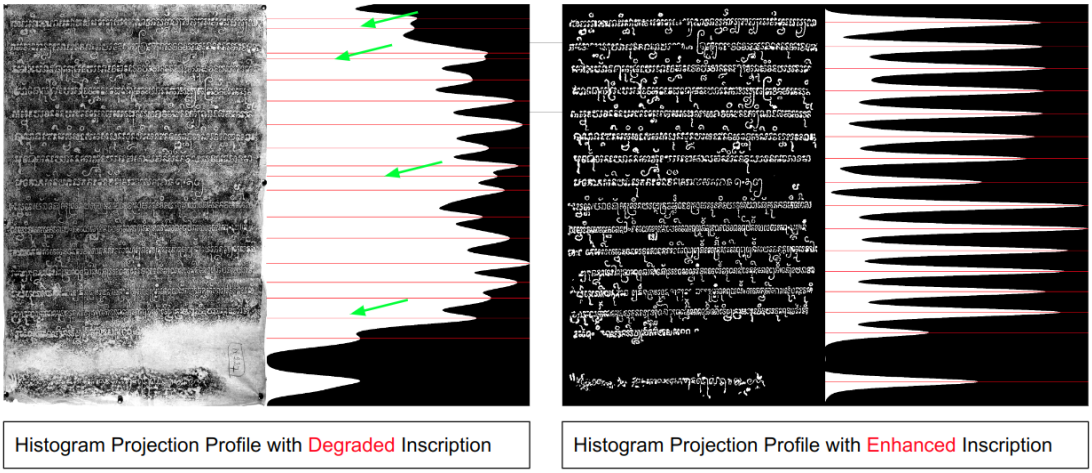


Figure 5.3: Comparison of the results after computation of the projection profile histogram on a degraded inscription and an enhanced inscription

to great variability in the representation of the characters and difficulties when analyzing heuristic model parameters. (See Fig. 5.4).

Figure 5.4: Examples of steles showing the evolution of the Cham writing style

- **Variability of the layout**: The position of the text is sometimes arbitrary. The spacing between two consecutive lines varies from one inscription to another. A lot of diacritics, ascenders, and descenders with positioning rules have to be considered. In addition, the

length of text lines varies a lot (some lines are very short while others are very long) which also leads to confusion when processing documents. Skew lines, lines that touch each other, and partially missing lines (due to damaged inscriptions) are problems that often appear in documents. Some examples of these issues can be seen in figure 5.5. In

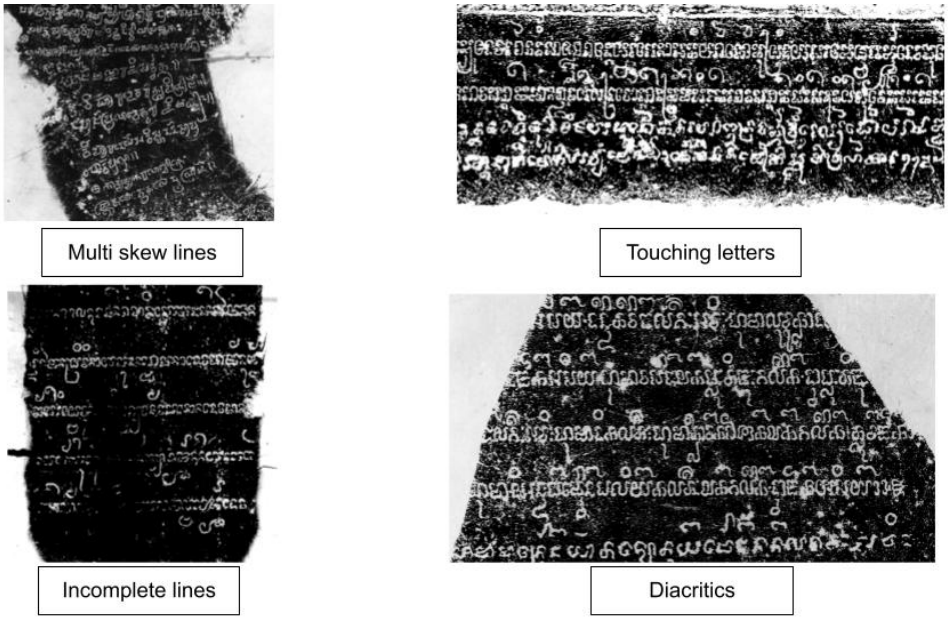


Figure 5.5: Examples of challenges for text line segmentation with Cham inscription images

addition, for Cham manuscript documents, we also come across several issues related to the structure of text lines such as incomplete lines or joining lines together.

As mentioned in the section 2.5.2, text line segmentation methods can work on both gray-scale and binary images. However, most of them still need a binarization step in some processes of the algorithm. In our work, the input images are binary images.

5.2 Improved Cost Function for Seam Carving based Approach (ICFSC)

Firstly, we present an overview of the proposed model for Cham documents, then an adaptation method for both Cham inscription and manuscript documents. While the baseline written in the Cham documents is quite straight, the most difficult of Cham documents is the complexity

of writing scripts which includes many diacritics, ascenders, and descenders. By observing the position of diacritics, ascenders, and descenders, we concluded that the different functions should be designed to take into account these conditions. For the detection of the baseline, an approach based on a histogram projection profile or using a pre-trained network can handle it. For diacritics, ascenders, and descenders, more information needs to be provided to correctly assign them. This kind of work can be done through the seam carving method Arvanitopoulos and Süsstrunk (2014). Seam carving for text line segmentation is based on the lowest calculated energy seam spanning the width of the image. The energy denotes the information about the text and non-text region (for example in Alberti et al. (2019), the higher value in the energy map represents the text area, and the lower value represents the non-text area). Thus, the lowest energy seam can be considered as the separation between text lines. However, the energy map proposed in Alberti et al. (2019), Arvanitopoulos and Süsstrunk (2014) is not sufficient to guide the seams correctly when working with Cham documents that have more accents or diacritics. This typical energy map does not allow these elements to be assigned to the correct line of text. Therefore, the idea is to add an appropriate cost function to improve the generation of seams. An overview of the proposed method is given in Fig. 5.6. The general proposed approach includes

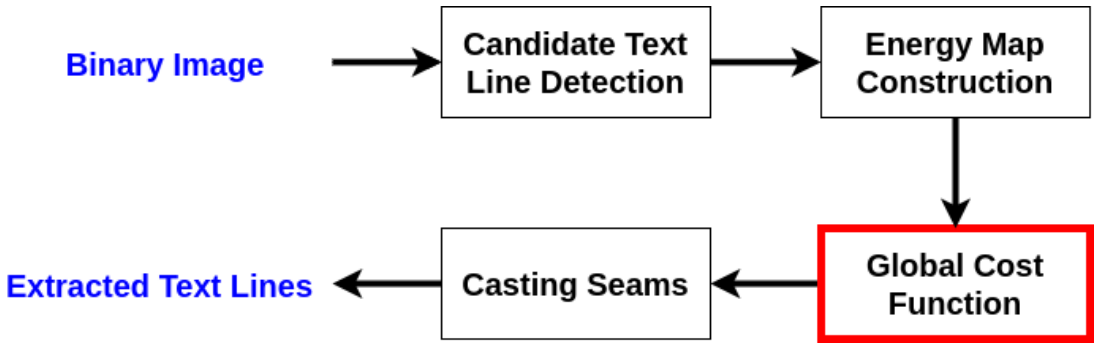


Figure 5.6: Overview of the proposed method

two phases. In the first phase, the candidate text line detection will be extracted through a histogram projection profile (for inscriptions) or pre-trained neural networks (for manuscripts). In the second phase, from each text line region, the separation seams will be generated by applying the seam carving method (energy map construction) combined with the proposed cost function (casting seams).

5.2.1 Candidate Text Line Detection

This step aims to detect the candidate lines from the input image. Firstly, let $I \in R_{h \times w}$ denotes the input image, and $\mathcal{L}$: $\{L_i\}$; $i = 1 : n$ denote the list of the candidate lines. Each candidate line L_i is represented by (l_i, m_i, u_i) where l_i, m_i, u_i represent respectively the ordinate of lower, middle, and upper line position of the candidate text line i^{th}. Depending on the type of document images, different methods have been used. Here, we present the histogram projection profile algorithm used on the **Textline-Inscription-Cham** and Diva-HisDB Simistira et al. (2017) dataset. For the **Textline-Manuscript-Cham**, we use a different approach for this step namely: the blob-extraction approach. The details of this method are presented in section 5.3.1. An illustration of the process on the Cham inscription document image is given in figure 5.7. First, the Sobel operator is applied to the input image to extract edges. The small components

Figure 5.7: Candidate line detection process : Candidate lines are shown in the most right image of figure. (Each candidate text line is in the region between two successive green lines.)

that can be considered as noise (still existing after the denoising step), are removed by comparing their areas with the average area of all the components in the image. We hence compute horizontal projection profile histogram H_I on the edge image. We found that the computation on the small columns Arvanitopoulos and Süsstrunk (2014) of the image does not work efficiently in the presence of a lot of diacritics, and descenders between two consecutive lines. So, this

projection profile is done on the whole image as below :

$$H_I = \sum_{i=0}^{h} I_i \tag{5.1}$$

where h is the height of the image. This histogram is smoothed by a cubic spline smoothing filter to reduce the number of false local maximums. This smoothing filter (f_σ) represents by parameter σ (smooth parameter) which is defined experimentally on the specific dataset.

$$H_{f_I} = f_\sigma(H_I) \tag{5.2}$$

One advantage of using a histogram on the whole image for Cham document images is that we can use the property space between lines as a heuristic rule to remove the close candidates. The peaks of the smoothed histogram are considered as the lower and upper position of the candidate line (green line on figure 5.7) while the valleys are considered as the middle position between two consecutive lines (red line on figure 5.7). The red lines will be used to provide more information when finding the most optimized path (separating seams).

5.2.2 Energy map construction

To cast the seams, an energy map needs to be constructed to guide the seams on how to separate the lines. Several methods can be used to compute the energy map including Gray-level Distance Transform (GDT) Levi and Montanari (1970), gradient map (GM) Arvanitopoulos and Süsstrunk (2014), or a combination of background energy and text energy (LCG) in Alberti et al. (2019).

Gray-level Distance Transform (GDT)

Gray-level Distance Transform (GDT) proposed in Levi and Montanari (1970) is computed as follows:

$$D(i,j) = \min\left\{d(i,j,k,l) + I(k,l)\right\}, \text{ for all } (k,l) \text{ in the image} \tag{5.3}$$

where $d(i, j, k, l)$ denotes the distance between pixels I(i, j) and I(k, l). In the implementation, we used d as Euclidean distance.

Gradient Map (GM)

Gradient Map (GM) presented in Arvanitopoulos and Süsstrunk (2014) is computed as follows:

$$E(i,j) = |\frac{I^{\sigma}_{i,j+1} - I^{\sigma}_{i,j-1}}{2}| + |\frac{I^{\sigma}_{i+1,j} - I^{\sigma}_{i-1,j}}{2}| \tag{5.4}$$

where I^{σ} is the smoothed input image I. In the implementation, since the input image is binary we don't apply smoothing on the image.

Labeling Cutting Grouping (LCG) Labeling Cutting Grouping (LCG) proposed in Alberti et al. (2019)

is computed as follows :

Background energy :

$$B(x_i) = \frac{1}{\underset{c \in CC}{min} ||l(x_1) - l(c)||} \tag{5.5}$$

where l(.) denotes the coordinates of the pixel; CC denotes the centroid of all Connected Components in the image; i=1, ... , $h \times w$.

Text energy :

$$T(B(x_i)) = \begin{cases} B(x_i) & if \quad x_i = 1 \\ 0 & if \quad x_i = 0 \end{cases}$$

Then the final energy S is the smoothed energy of text energy and background energy :

$$S(B(x_i), T(B(x_i))) = C_2(C_1(T(B(x_i) + B(x_i)))) \tag{5.6}$$

where C_2, C_1 are the convolution operations with kernels k_2 and k_1.

The visual comparison of the three methods is given in figure 5.8.

As we can see, the energy map proposed in Alberti et al. (2019) can clearly highlight the text parts while the two previous methods have almost the same value for baseline text and diacritics. The most important criterion why we chose the energy map proposed in Alberti et al. (2019) is based on the values energy map for the text of the baseline. This method can compute higher values for character parts and their neighborhoods. These values will be important for the next step of casting seams, this will prevent seams from jumping over the baseline of text lines since the Cham scripts have often spaces inside the characters. For this reason, we use the energy map calculation method detailed in Alberti et al. (2019) for our documents. Figure 5.9 visualizes the energy map with LCG and the position of the candidate lines.

5.2.3 Global cost function

Why do we need an additional cost function? We run a simple experiment with the use of the seam carving method with the calculation energy map of Alberti et al. (2019). Figure 5.10 illustrates some results obtained by the use of the energy map of Alberti et al. (2019). This figure shows that most of the diacritics are assigned to the wrong line. This energy map is not efficient enough to guide properly the seams. We, therefore, proposed an additional cost

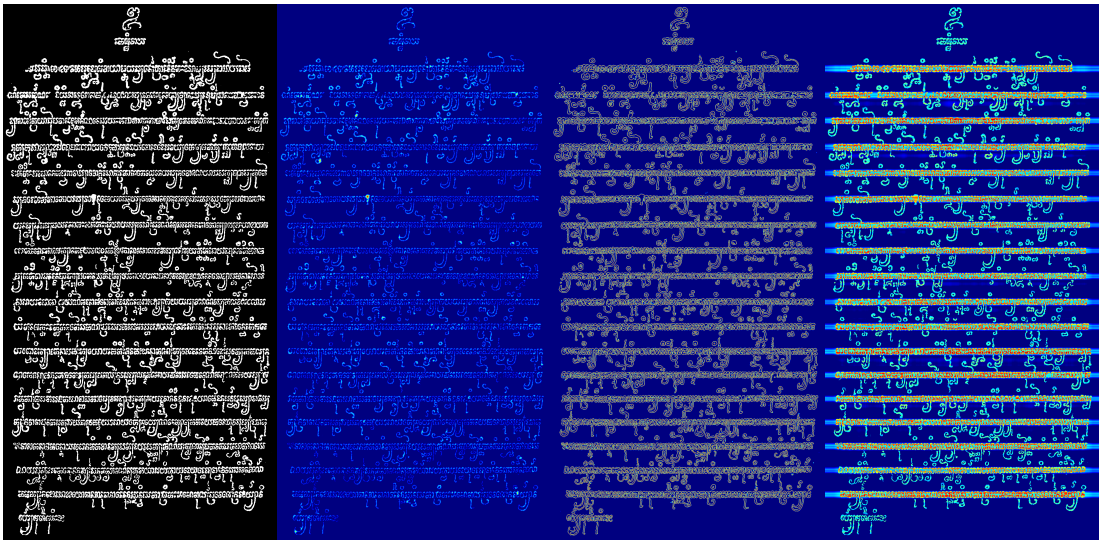


Figure 5.8: Energy map visualization of each method: (from left to right): input binary image, Distance Transform (Levi and Montanari (1970)), GM (Arvanitopoulos and Süsstrunk (2014)), LCG (Alberti et al. (2019)). The red color (resp. blue) represents high energy (resp. low)

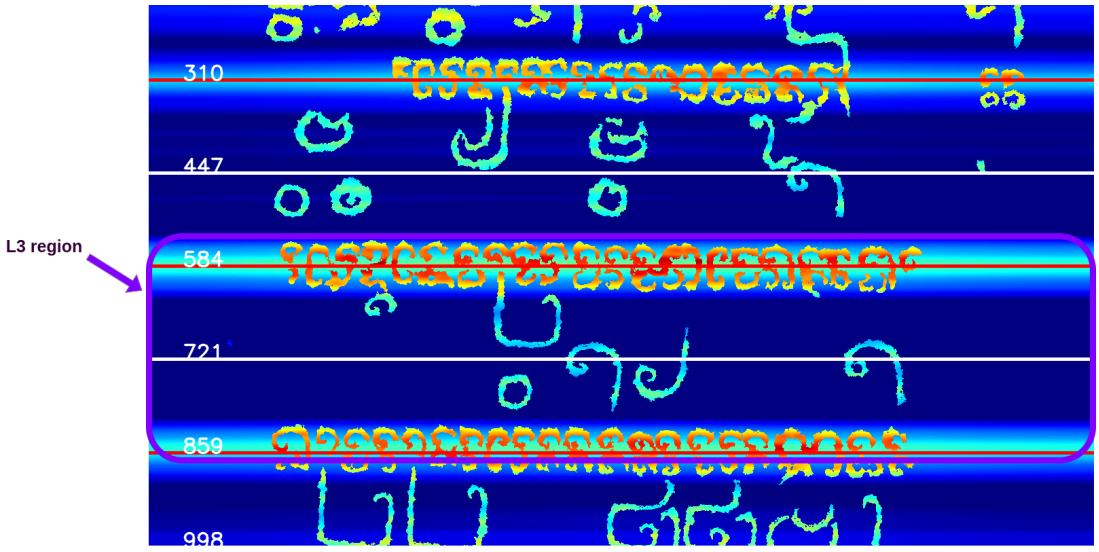


Figure 5.9: Heat maps of energy map computed with LCG Alberti et al. (2019). The red color (resp. blue) represents high energy (resp. low). Let's note that most of the characters in the main line are in red color. The red lines represent the peaks while the white lines represent the valleys of the smoothed histogram. One candidate line is represented by two consecutive peaks and one valley. For example, The candidate line L_3 is defined by (l_i, m_i, u_i) = (584, 721, 859).

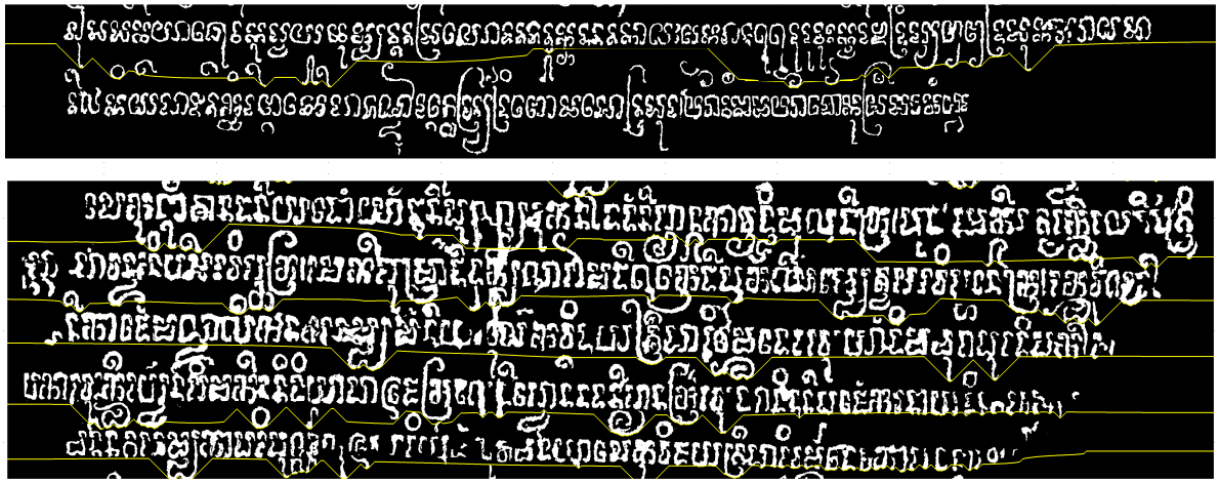


Figure 5.10: Separation lines (yellow lines) between text lines obtained with the simple seam carving method when using only energy map proposed in Alberti et al. (2019)

function when casting seams to handle this problem. This additional cost function is inspired by the cost function proposed in Surinta et al. (2014).
Our cost function is different from the work proposed in Surinta et al. (2014) by these three points.

1 - The global cost function consists of several components to consider the properties of text lines.

2 - We do not need to define either a starting point or a destination point for each component of the overall cost function.

3 - Our cost function is combined with the baseline energy map for casting seams.

Let's consider pixel p, its position (p_x, p_y), $V(p)$ the value of the pixel p and L_i the candidate line. The definition of the three components of our global function is presented in the following paragraphs :

Character cost function

The character cost function is defined by:

$$E_{C_p} = \begin{cases} 1 & \text{if, V(p) = 255} \\ 0 & \text{if, V(p) = 0} \end{cases}$$

The term simply forces the seams to avoid the character pixels when casting in the region.

Middle cost function

This term forces the seams to pass as close to the middle line and helps the seams return close to the middle line after moving away from it.

$$E_{M_p} = \begin{cases} \frac{p_y - m}{l - m} & if \quad p_y - m < 0 \\ \frac{p_y - m}{u - m} & if \quad p_y - m > 0 \end{cases}$$

where l, m, u are respectively the lower, middle, and upper ordinates of the candidate line which determines the region to which the pixel p belongs. p_y is the height ordinate of the point p. The value of this cost function reaches 0 when the point p has the same ordinate as the middle line.

Balance cost function

If we used only the character cost function, the seams, in some cases, can go up or down very close to character pixels. Moreover, in some cases where the middle line is not accurately determined, using only the **character cost function** described above may result in incorrect text line separation. Therefore, we proposed the **balance cost function** to encourage the seams to return to the **middle of gaps** between two lines. This term is computed as follows.
First, we denote L(p) and U(p) respectively the distance between the current point to the nearest white pixel, to the lower l_i and upper u_i ordinates of the candidate line L_i. The normalization of L(p) and U(p) is computed as:

$$\overline{L(p)} = \frac{L(p)}{L(p) + U(p)} \tag{5.7}$$

$$\overline{U(p)} = \frac{U(p)}{L(p) + U(p)} \tag{5.8}$$

Then the balance cost function is evaluated by:

$$E_{B_p} = \begin{cases} \frac{1}{\overline{U(p)} * \overline{L(p)}} & if \quad \frac{\overline{U(p)}}{\overline{L(p)}} < 8 \quad \text{or} \ \frac{\overline{L(p)}}{\overline{U(p)}} < 8 \\ 10 \quad otherwise \end{cases}$$

We set the limit value of the ratio $\frac{\overline{U(p)}}{\overline{L(p)}}$ and $\frac{\overline{L(p)}}{\overline{U(p)}}$ to 8 because when this ratio is greater than 8 the cost function will reach too big value (for example if the two values $\overline{U(p)}$ and $\overline{L(p)}$ are 0.89 and 0.11 this cost is around 10 but for 0.95 and 0.05 the cost function is equal to 21). This term forces the seams to pass in the **middle of gap** between two white pixels, to avoid jumping to the previous or next line. Experimentally, we observe that the ideal values of this function are between 4 and 10. The smallest score is obtained when the seams are located in the **middle of gaps**.

Global cost function

Finally, the global cost function is a weighted sum of the three components detailed above:

$$C_p = w_1 * E_{C_p} + w_2 * E_{B_p} + w_3 * E_{M_p} \tag{5.9}$$

This global cost function allows to balance of the three components to better guide casting seams. An illustration of each component of the global cost function is given in Fig. 5.11.

5.2.4 Casting seams

In this step, we generate the separate seams by using the baseline energy map and the global cost function.

Let's denote the space from l_i to u_i (corresponding to the lower and upper ordinate of the candidate line L_i), $E_{l_i:u_i,w}$ the energy map and $C_{l_i:u_i,w}$ the global cost function where w is the width of the image.

Seam carving method consists of two steps: a forward step and a backward step.

- In the forward step, the seam is initialized from each vertical pixel from l_i to u_i. Each seam moves from the current column to the next column by computing the scores $M_{y,x}$ and $M_{y,x+1}$ as follows :

  At the first column :

  $$M_{y,1} = E_{y,1} \tag{5.10}$$

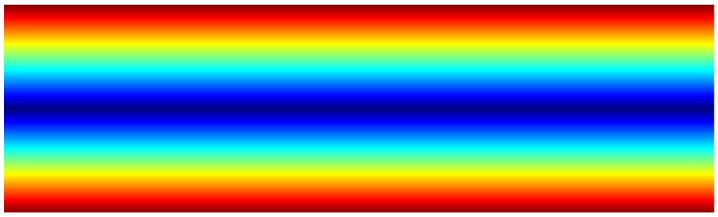
  then for the next columns:

  $$M_{y,x+1} = E_{y,x+1} + \alpha * C_{y,x+1} + \min(M_{y-1:y+1,x-1}) \tag{5.11}$$

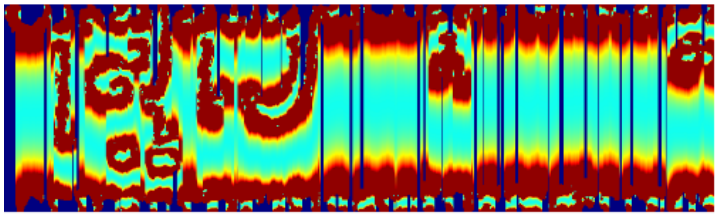
  with $y \in [l_i, u_i]$ and $x \in [2, w]$; Since the range of the energy map and the global cost function is different, so we used α as the weight on the global cost function to balance

(a) Cropped region from candidate line

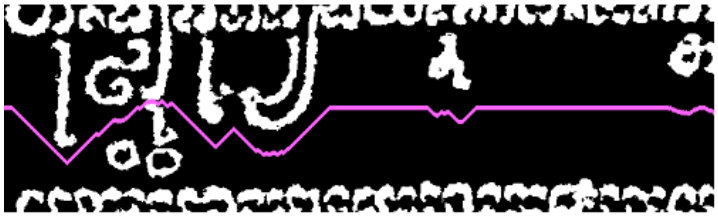
(b) Character cost function heat map



(c) Middle cost function heat map



(d) Balance cost function heat map



(e) Separation seam

Figure 5.11: Heat map of each component of the global cost function. From top to bottom: input image, Character cost, Middle cost, and Balance cost functions; Separation seams (in purple). For the character cost function, high values (red) correspond to character pixels while low values (blue) correspond to the background. In the middle cost function, pixels close to the middle line are in blue and represent lower values, values increase (cyan, green, yellow, and red) for pixels moving away from the middle line. For the last cost function, the middle gaps between two lines have the lowest values (cyan) and the values increase (red) when getting closer to the previous or the next line.

them. This process is sequentially done from the first column on the left to the last column on the right of the image.

- In the backward step, we construct the separating seam from the position with the lowest cumulative score of $M_{l_i:u_i,w}$ then this process is repeated from right to left by choosing the pixel with the lowest cost on the left column. To create a continuous separation line, only the adjacent pixels are considered (pixels at the previous, current, and next line).

5.2.5 Post processing

As results from casting seams are not always properly assigned some diacritics so an additional post-processing is applied to enhance the segmentation. In the literature, several post-processing approaches are proposed such as fast approximate energy minimization in Barakat et al. (2021a), Minimum Spanning Tree (MST) in Held and Karp (1971) and Alberti et al. (2019).

In this work, we propose to use Minimum Spanning Tree as post-processing method. Minimum Spanning Tree is a subset of the edges of a connected, edge-weighted un-directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. An illustration of Minimum Spanning Tree is given in the figure 5.12. The use of MST

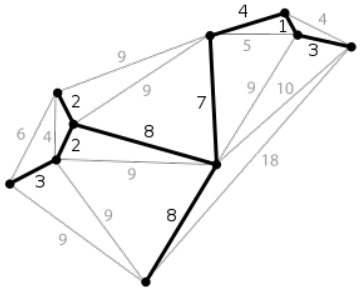


Figure 5.12: Example of Minimum Spanning Tree on the simple graph. This minimum spanning is the collection of bold connections in the graph whose sum of edge weights is the smallest.

is detailed in the algorithms 4:

Moreover, using MST, we can create a tight polygon around the line instead of just taking the open area around the line. A visualization of the use of MST on Cham manuscript documents is

Algorithm 4 The use of MST on text line

Require: The list of upper and lower polygons extracted from casting seams.
Take the image inside the lower and upper polygon pair
Get all connected components in the extracted image
Build a graph from the defined representation points for each connected component
Extract polygons ← Finding the optimal link from graph

provided in figure 5.13.

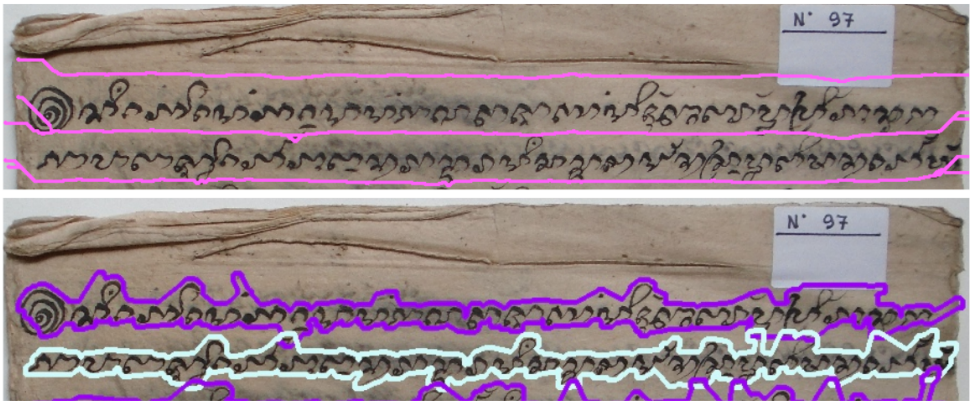


Figure 5.13: Visualization of text line segmentation with MST post-processing. Top : image seams generated from casting seam; bottom : Results after application of the MST.

5.3 Configuration

5.3.1 Data preparation

The proposed text line segmentation approach has been evaluated on several datasets. Each of them is described in the following paragraphs.

Textline-Inscription-Cham

Instead of using the original inscription images that are highly degraded, we used the trained model described in the image enhancement chapter 4 for pre-processing the whole images. After that, the enhanced images are re-cleaned and then segmented manually line by line by our linguistic expert. We have segmented 26 inscription images and obtained 395 text lines.

Textline-Manuscript-Cham

In this section, we detail the process to create the Cham manuscripts dataset. The general principle as proposed above is still applied. However, we use different methods for each step. The reason for this change comes from the characteristics of Cham manuscript documents. This dataset has a layout quite close to other datasets such as READ-BAD Grüning et al. (2018), DIVA-HisDB Simistira et al. (2017), Balinese palm leaf manuscripts Kesiman (2018), which are well-studied in the literature. So, better results are obtained if we can take advantage of an available pre-trained model.

Specifically, we replace the histogram projection profile with the blob extraction strategy where the blobs are extracted from the pre-trained deep learning model. Here, we use the pre-trained model proposed in Monnier and Aubry (2020) to extract these blobs. The model in Monnier and Aubry (2020) is based on the work of Oliveira et al. (2018), in which an encoder-decoder model with backbone Resnet-50 with the additional post-processing is used to generate the blobs (baseline).

Furthermore, the input binary images are not available, so the binarization step is done through the pre-trained model. We have adapted the pre-trained model of Calvo-Zaragoza and Gallego (2019) for image binarization whose architecture is based on convolutional encoder-decoder. This model is trained on ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016) dataset Pratikakis et al. (2016). As shown in the figure 5.14, after the binarization step and the blob extraction respectively from pre-trained models Calvo-Zaragoza and Gallego (2019) and Monnier and Aubry (2020), there still exists some unwanted parts such as notation on the manuscripts, partially extracted parts, stamps. So, more processing image has been used to handle it. We first eliminate these "noisy" elements by checking the connected components whose area is less than the average area. Another issue that appears is the big connected components generated by merging two consecutive lines as in figure 5.15. This situation provides wrong candidate text lines so some adjustments are necessary to fix these connected components. These connected components are considered to be split if the conditions about their height and area are not satisfied.

The details of the condition to split these components are given below in Algorithm 5.

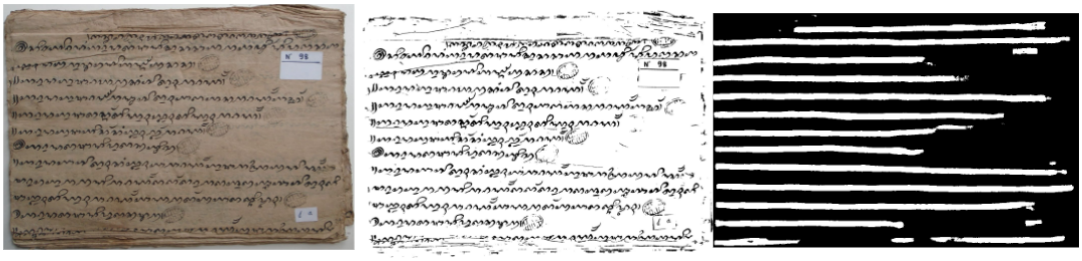


Figure 5.14: Results of the binarization (center) and blob extraction (right)

Figure 5.15: Example of merging between two consecutive lines. (Left) Original image, (middle) The red circle shows the connected components in contact. (Right) Result after splitting process.

Algorithm 5 Splitting of the big connected components

Require: List of all connected components cc^1, cc^2, ..., cc^L
  if $cc^i_h \leq T_{height}$ **then**
    continue
  else
    $f \leftarrow POLYFIT(x, y)$ {x,y is the coordinate of all points in the connected components cc^i, POLYFIT(x,y) is a linear function (polynomial fitting) that approximate of all points in the connected components cc^i. It is calculated by minimizing least squares polynomial errors.}
    $\hat{y} \leftarrow f(x)$
    $err_{distance} \leftarrow MSE(\hat{y}, y)$ {MSE is the mean square error.}
    if $err_{distance} \geq T_{dist}$ **then**
      SPLIT cc^i
    end if
  end if

Furthermore, the bad blob generation can cut one text line in several connected components. It is therefore necessary to reconnect the different parts of the same text line. By analyzing these problems, two components can be considered as part of one text line if the average of height coordinates between two components is less than T_{cch} and the angle created by two components is less than T_{cca}. Before merging, a skeletonization transformation is applied to easily facilitate the operation. The checking condition algorithm is given below in Algorithm 6.

Algorithm 6 Checking condition algorithm for merging connected components

Require: Two connected components cc^1, cc^2 of skeleton image.
  $avgh_{cc1} = AVERAGE(cc_h^1)$ {cc_h^1 is height coordinate of all points in the cc^1}
  $avgh_{cc2} = AVERAGE(cc_h^2)$ {cc_h^2 is height coordinate of all points in the cc^2}
  $distance \leftarrow ABS(avgh_{cc1}, avgh_{cc2})$
  if $distance \leq T_{cch}$ **then**
    continue
  else
    $angle_1 \leftarrow ANGLE(cc^1)$ {$ANGLE(cc^1)$ is the angle created by all points in connected component cc^1}
    $angle_2 \leftarrow ANGLE(cc^2)$ {$ANGLE(cc^2)$ is the angle created by all points in connected component cc^2}
    $diff_{angle} \leftarrow ABS(angle_1, angle_2)$
    if $diff_{angle} \leq T_{cca}$ **then**
      Merging
    else
      $f \leftarrow POLYFIT(x, y)$ {(x, y) is the coordinate of all points of $cc^1 \cup cc^2$, POLYFIT is defined the same in algorithm 5}
      $\hat{y} \leftarrow f(x)$
      $err_{distance} \leftarrow MSE(\hat{y}, y)$ {MSE is the mean square error}
      if $err_{distance} \leq T_{err}$ **then**
        Continue
      else
        Merging
      end if
    end if
  end if

After splitting and merging connected components, the remaining components are considered as the candidate lines and the same text line segmentation step used in Cham inscription will be applied. An example of the complete workflow for processing manuscripts is shown in figure 5.16. As no ground truth is available for **Textline-Manuscript-Cham**, we only evaluated the results qualitatively.

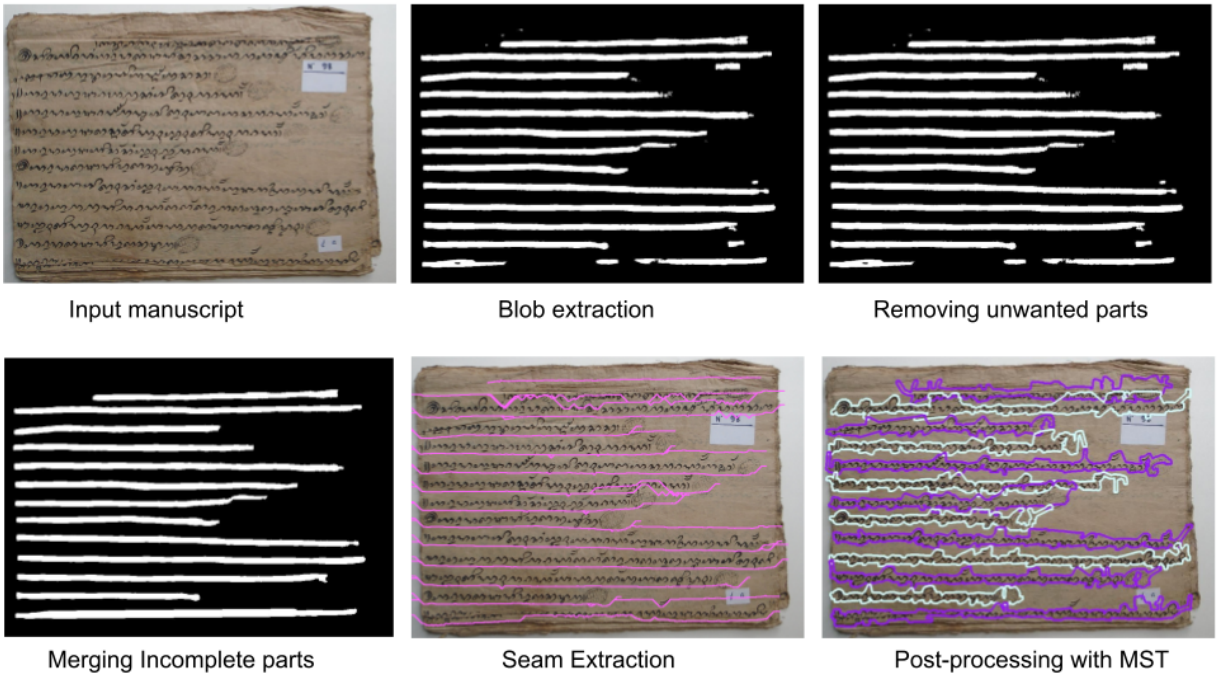


Figure 5.16: An illustration text line segmentation step applying for manuscripts collection

Diva-HisDB dataset Simistira et al. (2017)

DIVA-HisDB is a public historical dataset for the evaluation of several DIA tasks such as: layout analysis, text line segmentation, binarization, and writer identification. DIVA-HisDB dataset Simistira et al. (2017) contains 150 pages of three different medieval manuscripts (CB55, CSG863, CSG18) dating from 11^{th} to the 14^{th} century and digitized with a resolution of 600 dpi camera. The documents have a size of approximately 20x25 cm. The collection consists of a complex layout containing the main text body, interlinear glosses, additions, and corrections. Ground truths are provided at both pixel and text levels. The particularity of this dataset includes interlinear glosses, lettrines decoration (drop caps), and scripts of different sizes. As with some works of the literature on this dataset, we use the good binarization provided with the dataset as input for our algorithm. For a fair evaluation, we use the same private test set of the ICDAR 2017 competition Simistira et al. (2017). This test set is constituted of 30 pages (875 lines). For candidate line detection, the same process used for inscription collection is applied.

5.3.2 Parameters Settings

In this section, we define a set of parameters for each dataset we used in this task. First, the shared configuration between each dataset is presented. The α weight of the global cost function with energy map is set to 0.5.

For **Textline-Inscription-Cham** dataset, we set σ (smoothing filter) to $1e-6$, three weights of the global cost function w_1, w_2, and w_3 (character cost function, middle cost function, and balance cost function) are set experimentally respectively to 10, 3, and 0.02.

For the Diva-HisDB dataset, we set σ to $5e-4$, three weights of the global cost function w_1, w_2, and w_3 set to 10, 1, and 0.01.

For **Textline-Manuscript-Cham** dataset, we set three weights of the global cost function w_1, w_2, and w_3 are 5, 3, and 0. (We do not use projection profiles). Related to some parameters in the post-processing step, we set the value of T_{height}, T_{dist}, T_{cch}, T_{cca} and T_{err} are 65, 80, 21, 5 and 5.

5.4 Metrics and Evaluation

5.4.1 Metrics

To evaluate the results of the experiments and facilitate the comparison with other approaches, we use two sets of metrics : F-Measure and Line IU / Pixel IU. We could also use F-Measure for DIVA-hisDB, but for a fair comparison with the other methods, we have preferred to use the same metrics used in the literature: Line IU / Pixel IU. For **Textline-Inscription-Cham** dataset, we have a ground truth at the line level so we have adapted the F-Measure. F-Measure (FM) is used from Stamatopoulos et al. (2013) which computes from Detection Rate (DR), Recognition Accuracy (RA). This metric is used to evaluate the text line segmentation on **Textline-Inscription-Cham** dataset. The F-Measure is calculated as follows:

Detection Rate (DR):

$$DR = \frac{M}{N_1} \tag{5.12}$$

Recognition Accuracy (RA):

$$RA = \frac{M}{N_2} \tag{5.13}$$

F-Measure (FM)):

$$FM = \frac{2 \times DR \times RA}{DR + RA} \tag{5.14}$$

where N_1 and N_2 are respectively the number of lines in ground truth and the number of segmented lines. M is the number of one-to-one matches. Matches are obtained when the intersection region between the segmented and ground truth line is greater than a threshold λ. λ is defined experimentally and set to 90 on **Textline-Inscription-Cham** dataset.

For the Diva-HisBD dataset, we have adopted the metrics Line IU and Pixel IU Simistira et al. (2017) which use in different papers of the literature.

Pixel IU is defined by:

$$\text{Pixel IU} = \frac{TP}{TP + FP + FN} \tag{5.15}$$

TP is the number of pixels that are correctly detected, FP is the number of extra pixels and FN is the number of missed pixels. All the metrics take into account the only foreground pixels.

Line IU is defined by:

$$\text{Line IU} = \frac{CL}{CL + ML + EL} \tag{5.16}$$

where CL is the number of lines detected correctly, ML is the number of missed lines and EL is the number of extra lines. Lines are considered correct when line precision and line recall are above a given threshold. For a fair comparison, we used the same setup evaluation as the one used in the ICDAR2017 Competition on Layout Analysis for Challenging Medieval Manuscripts Simistira et al. (2017), with a threshold value of 75. For **Textline-Manuscript-Cham** dataset, since the ground truth is not available, so we have only the qualitative results.

5.4.2 Experiments

In this section, we present several experiments on different datasets to evaluate the performance of the proposed method :

- On the **Textline-Inscription-Cham**, we made two experiments. The first aims to evaluate the contribution of different modules of the proposed method (ablation study). The second is a comparison with other methods in the literature.
- On the **Textline-Manuscript-Cham**, as no ground truth was available, we propose a qualitative evaluation of the proposed method with two off-the-shelf methods.
- Finally, the generalization of the proposed method is tested on the Diva-HisDB dataset.

Textline-Inscription-Cham

Ablation study : The first experiment aims to evaluate the role of each component of the proposed method. First, we evaluated results when using only the energy map (cf. Section 5.2.2), then only the global cost function (cf. Section 5.2.3), and finally the proposed method which combines the energy map and global cost function. The obtained results for **Textline-Inscription-Cham** dataset are shown in Tab. 5.1.

Table 5.1: Ablation study on the **Textline-Inscription-Cham**

	DR (%)	**RA (%)**	**FM (%)**
Only energy map	80.76	81.58	81.17
Only global cost function	90.37	91.30	90.84
The proposed method without Balance cost	90.63	91.56	91.09
The proposed method (ICFSC)	**91.14**	**92.10**	**91.60**

We observe that the use of the global cost function (second row in the table) significantly improves the results from around 80% to 90% for all metrics. Moreover, the use of the energy map of Alberti et al. (2019) also further improves results (around 1%). So, a combination between the energy map and the global cost function is necessary.

The weights of each component of the global cost function are important elements whose impact must be analyzed. The weight of the **Balance cost function** is quite small in comparison with the ones of the **Middle cost function** and the **Character cost function**. The impact of the **balance cost function** was studied by evaluating the proposed method with or without this cost

function. Experimental results show that thanks to the **balance cost function**, the values of DR, RA, and FM are improved from 90.63%, 91.56%, and 91.09% to 91.14%, 92.10%, and 91.60% respectively. The qualitative effects of the balance cost function are shown in figure 5.17. By using this **balance cost function**, the separation seams can move up the top (correct segmentation) instead of moving down (wrong segmentation).

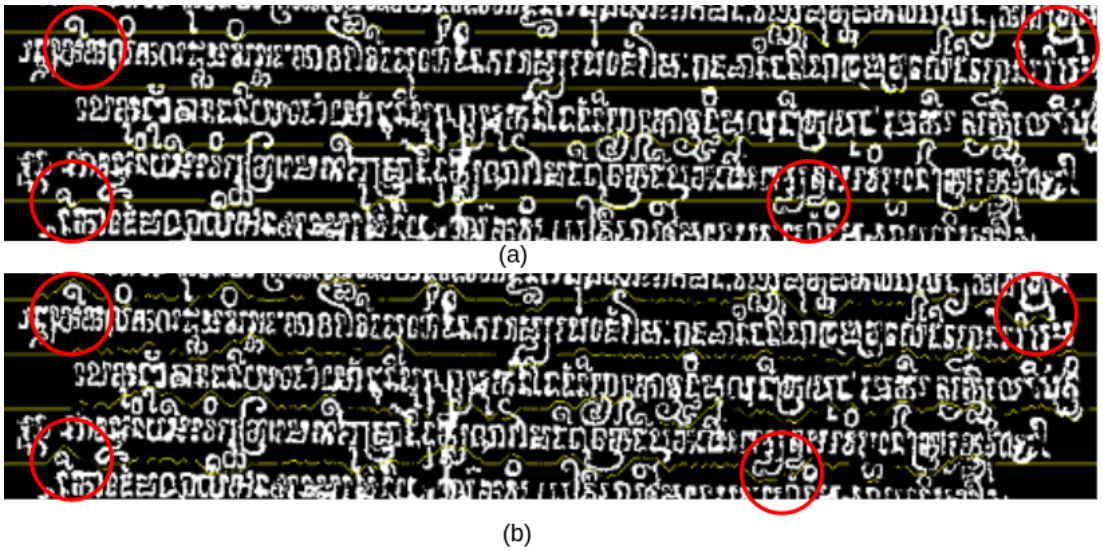


Figure 5.17: Text line segmentation results without (a) / with (b) balance cost function. The red circles indicate the areas where the results have been improved.

Comparison with other methods of the literature : In the second experiment, we compared the results of the proposed method with three baseline approaches that are seam-carving-based Arvanitopoulos and Süsstrunk (2014), A* path planning Surinta et al. (2014) and LCG Alberti et al. (2019). The results are shown in Table 5.2. On **Textline-Inscription-Cham** dataset,

Table 5.2: Comparison with the state-of-the-art methods on **Textline-Inscription-Cham** dataset

	DR	**RA**	**FM**
Seam-carving based Arvanitopoulos and Süsstrunk (2014)	54.68	52.17	53.40
LCG Alberti et al. (2019)	65.31	72.47	68.70
A* path planning Surinta et al. (2014)	84.8	85.71	85.25
The proposed method (ICFSC)	**91.14**	**92.10**	**91.60**

the proposed method obtained 91.14%, 92.10%, and 91.60% for DR, RA, and FM metrics, it outperforms all the studied methods for all metrics. A qualitative comparison between A* path

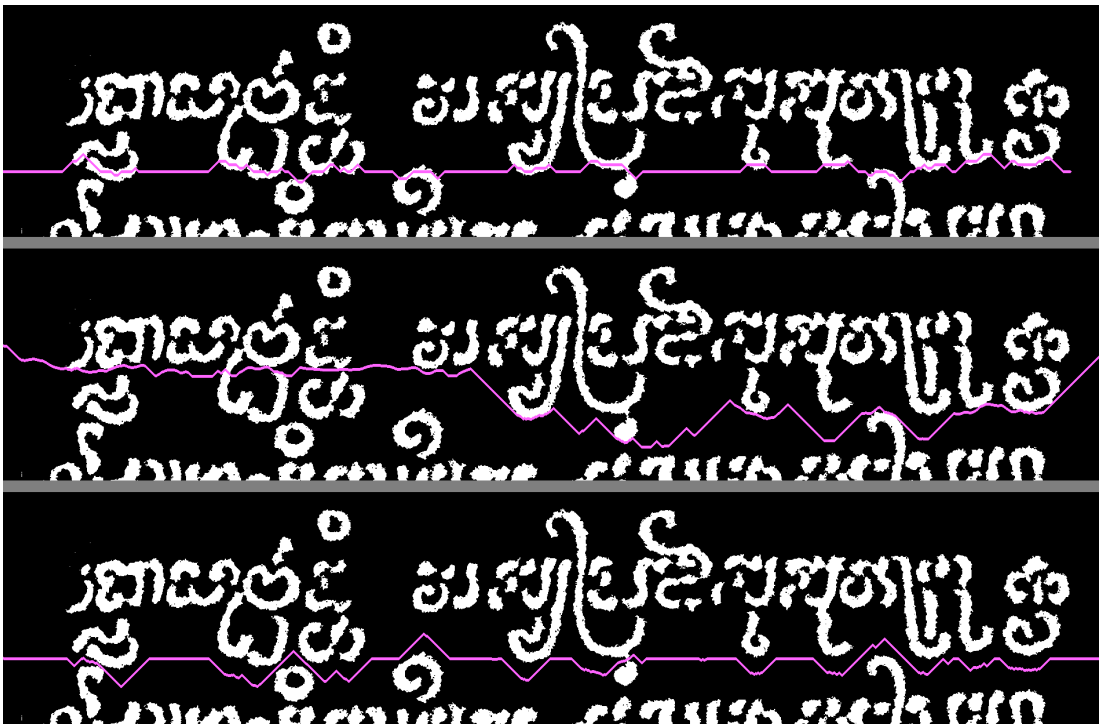


Figure 5.18: Qualitative results on **Textline-Inscription-Cham** dataset. From top to bottom: A*path planning Surinta et al. (2014), seam carving based Arvanitopoulos and Süsstrunk (2014), proposed method. Separation seams are in purple.

planning, seam carving-based approach, and the proposed method is shown in figure 5.18. We observe that with the proposed method, separating seams are more flexible and can avoid the characters when going back to the middle line by choosing empty spaces between the two lines.

Textline-Manuscript-Cham

As the ground truth for the **Textline-Manuscript-Cham** dataset is not available, only qualitative results are reported here. The **Textline-Manuscript-Cham** dataset has been applied on the two systems eScriptorium and Transkribus (See section 2.7) whose segmentations are based on the work of Kiessling (2020) and Grüning et al. (2019)), respectively. The first analysis aims to evaluate the ability to detect the baseline (blob extraction) of the documents. The figure 5.19) presents the results obtained respectively with Transkribus, eScriptorium, and the proposed method. We observe that all the methods correctly detect the baselines of the documents

in the general case where they contain mainly straight lines. However, there are still some types

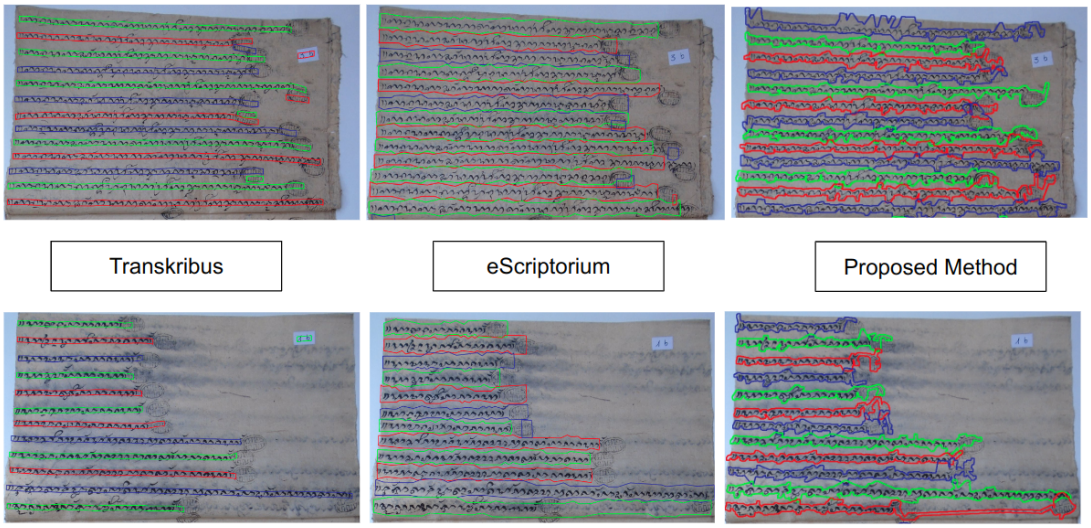


Figure 5.19: Baseline detection with different methods

of documents where the segmentation of baseline is not correct. These incorrect results appear in two cases: ink-bleed through (See Fig. 5.20) and short lines (See Fig. 5.21). Considering

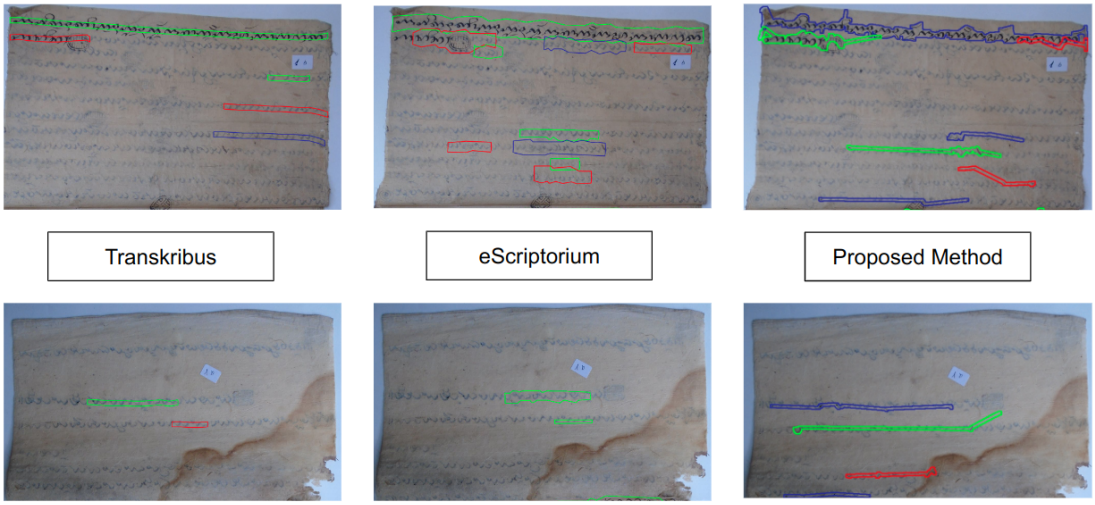


Figure 5.20: Ink-Bleed Through issues on manuscripts

the baseline segmentation, on all studied **Textline-Manuscript-Cham** dataset, the pre-trained model of Transkibus gives the best results by minimizing the issues described above. In order to obtain better results on this dataset, a fine-tuning of these pre-trained models are necessary.

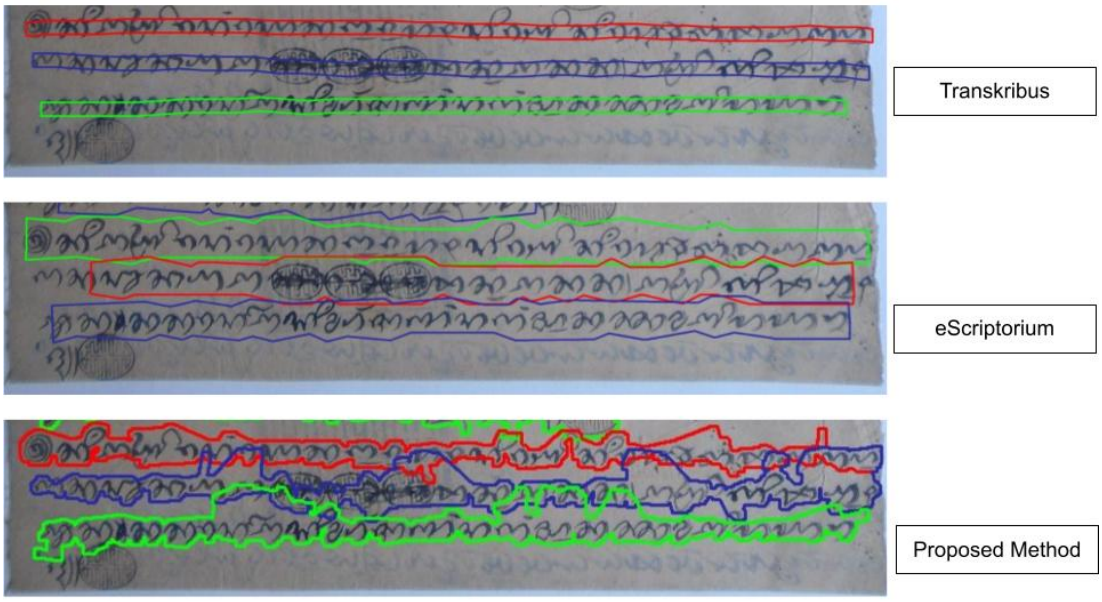


Figure 5.21: Short lines issues on manuscripts

After analyzing the baseline segmentation, we have considered the segmentation of the whole lines because this task is very important for the next step: text recognition/transliteration. In this analysis, only the results of eScriptorium and the proposed method are compared since the Transkribus provides only for the baseline detection. The first observation is that the segmentation with eScriptorium often misses certain parts of the lines such as the beginning or the end of the line (See Fig. 5.22). Moreover, many lines are segmented into multiple segments (See Fig. 5.23). By using merging connected components, the proposed method can correctly segment these text lines. We also observe that for the line whose writing is non-homogeneous (See Fig. 5.24), the proposed method can handle this while eScriptorium can not.

We also observed that the proposed method sometimes generates an over-segmentation. This happens when the distance between the lines is quite small or the dilation of the blob image overlaps another line. The limit tends to cover some part of the previous or next lines (See Fig. 5.25). This behavior is acceptable because it still retains enough information for recognition. However, additional experiments should be necessary to evaluate the impacts of these over-segmentations. We think that these kinds of issues can be solved by getting a better pre-trained model in the extraction of blob and binarization results.

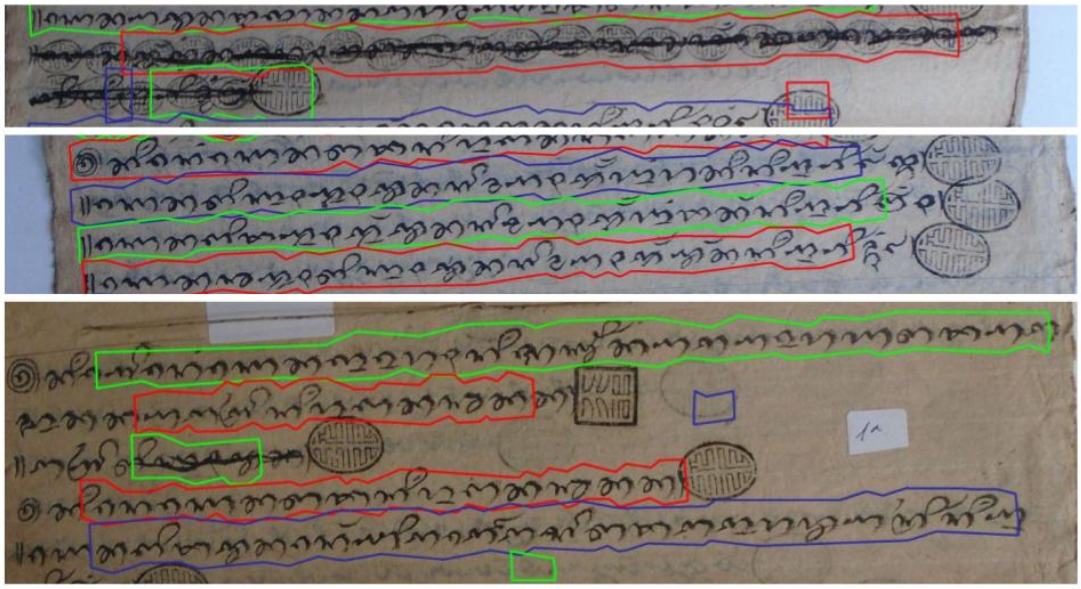


Figure 5.22: Missing segmentation on **Textline-Manuscript-Cham** dataset with eScriptorum

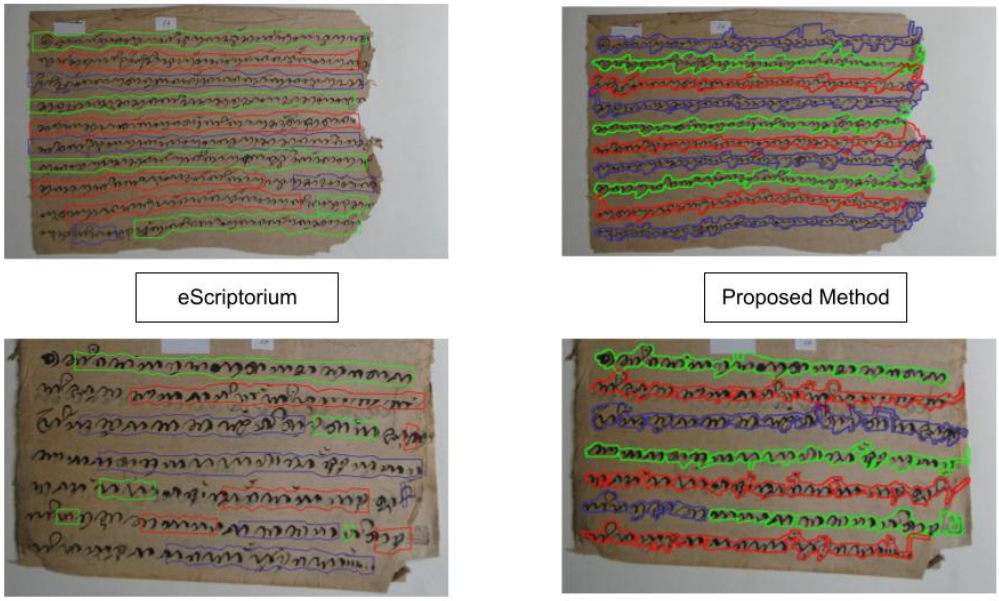


Figure 5.23: Text line segmentation on **Textline-Manuscript-Cham** dataset in the case of splitting into multiple segments with eScriptorum

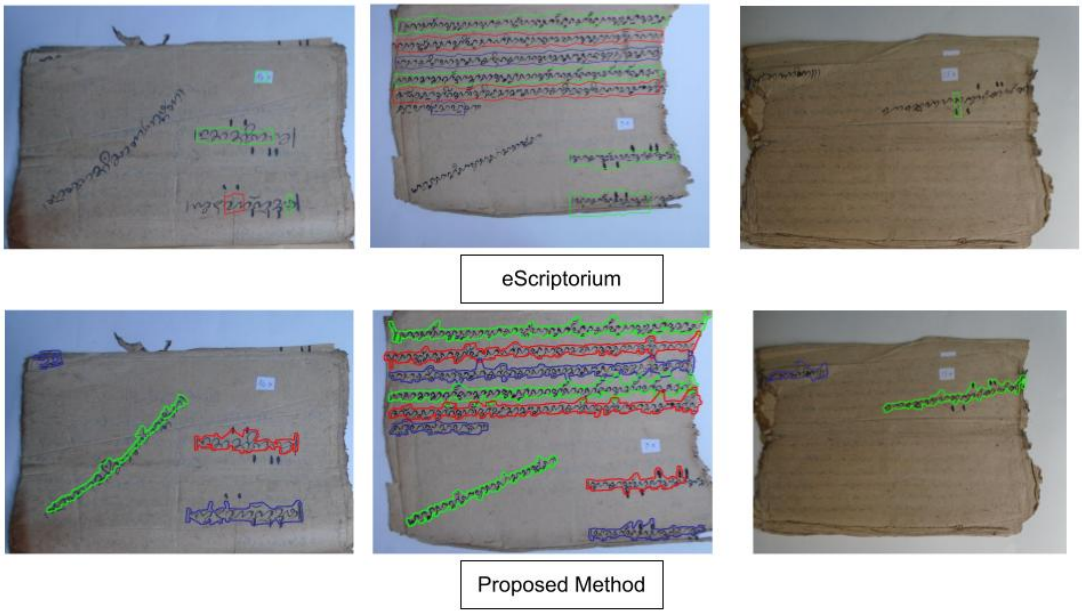


Figure 5.24: Text line segmentation performance in case of non-homogeneous lines.

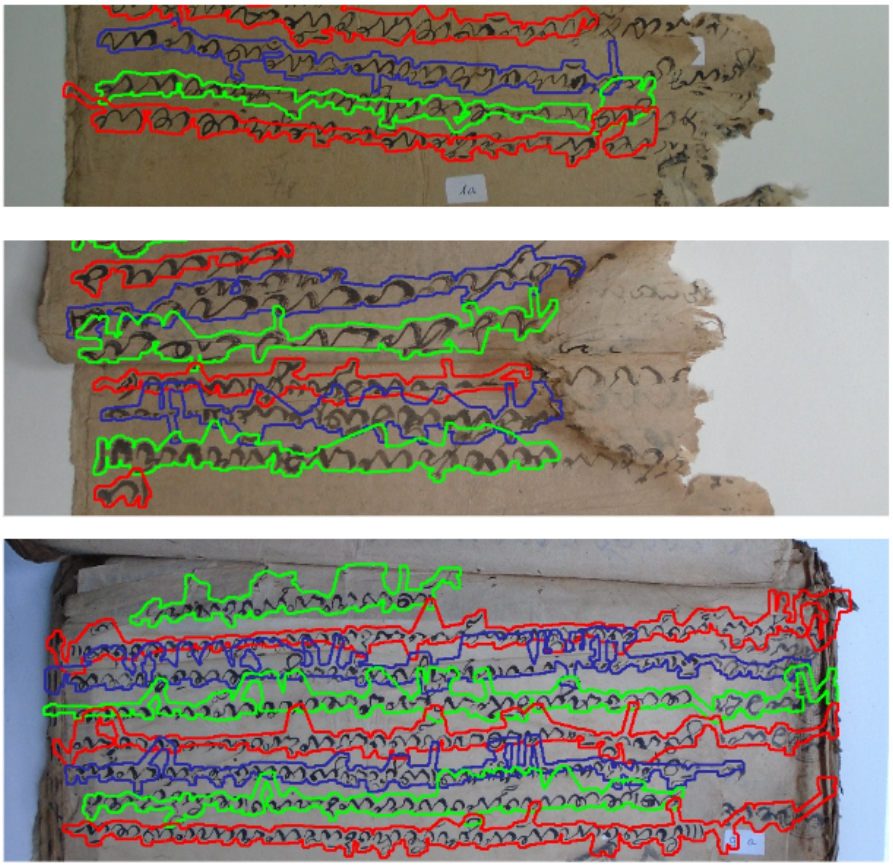


Figure 5.25: Over-segmentation with the proposed method

We have also figured out two main cases where both models could not produce good results :

- when a line has to split into two lines (See Fig. 5.26) even if it corresponds to the same writing line.
- when the line starts with a "signature" (represented by a small straight line in figure 5.27).

However, there are still cases that are so tough that it is difficult to find a model capable of dealing with them.

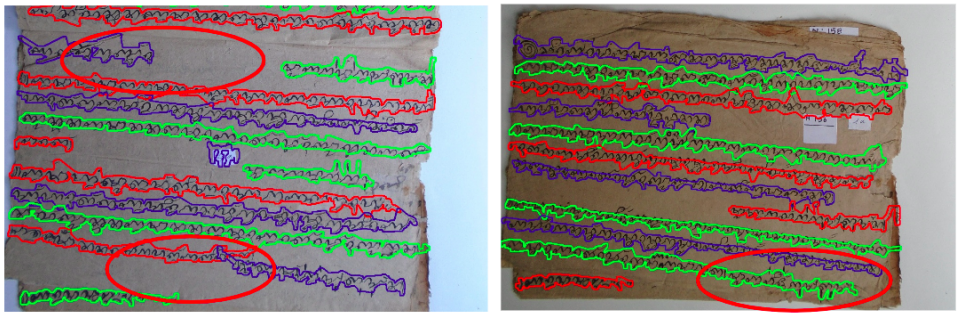


Figure 5.26: Cases where the lines are considered to split into two segments. (Left) correct segmentation, the lines are split into two distinct lines (at the red circle). (Right) wrong segmentation, the line has to be split into two lines.

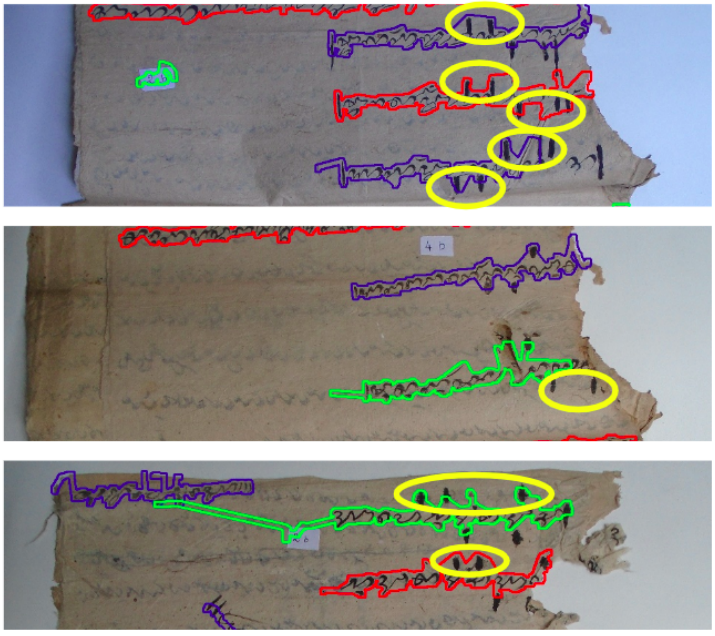


Figure 5.27: Incorrect segmentation with signature (Yellow cirles).

Diva-HisDB dataset

In this section, we evaluate the robustness of the proposed method on the benchmark Diva-HisDB dataset. The results are given in table 5.3. We can note that in comparison with seam

Table 5.3: Comparison results on the DIVA-HisDB dataset

	Line IU	**Pixel IU**
Wavelength (Seam Carving based) Seuret et al. (2017)	97.86	97.05
LCG Alberti et al. (2019)	**99.42**	96.76
FCN+EM Barakat et al. (2021a)	99.21	97.53
UTLS Barakat et al. (2020)	97.85	97.04
Seam carving with only cost function	99.36	96.88
The proposed method (ICFSC)	**99.36**	**98.86**

carving based method Seuret et al. (2017), our approach improved both Line IU and Pixel IU scores. The comparison with deep learning approach Barakat et al. (2021a), Barakat et al. (2020), shows that our method obtains competitive results for Line IU score and the highest result for the Pixel IU score. Figure 5.28 illustrates some qualitative results. We observed that the proposed method can avoid most of the ascending and descending parts of the characters.

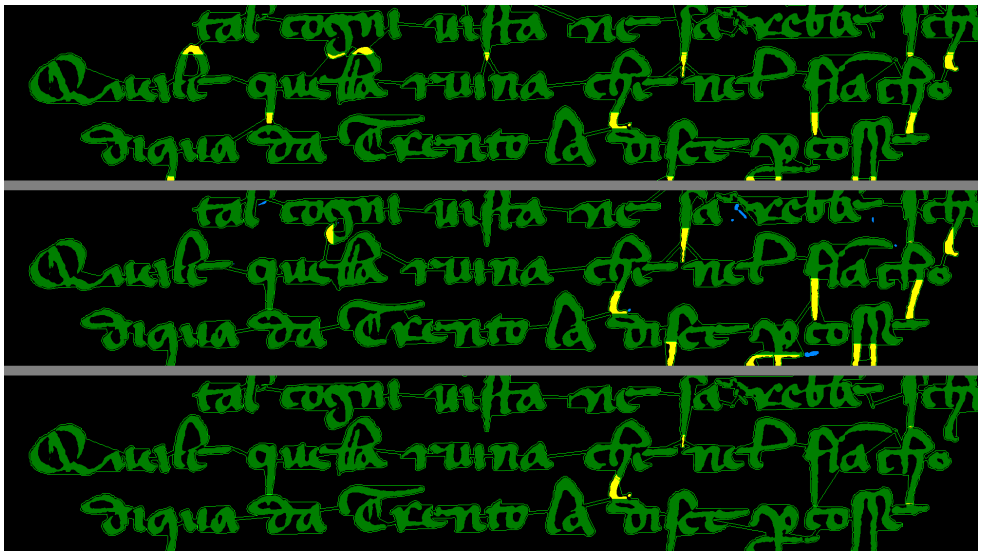


Figure 5.28: Qualitative results on DIVA-HisDB. From top to bottom: Seam carving method with the only global cost function, LCG Alberti et al. (2019), the proposed method. The green parts correspond to the correct segmentation of foreground pixels while the yellow parts correspond to the wrong segmentation.

5.5 Conclusions

In this section, we present the text line segmentation problem on Cham documents. Considering the characteristics of the Cham writing system, we propose a general approach for text-line segmentation which is based on the seam carving method. To more accurately assign the diacritics, ascenders, and descenders to the good text line, a global cost function consisting of several components has been proposed. From the proposed general approach, different techniques have been used to adapt to different datasets.

The effectiveness of the proposed method is evaluated on three different datasets. On the benchmark Diva-HisDB Simistira et al. (2017) dataset, the proposed method obtains competitive results compared to deep learning methods. For the **Textline-Inscription-Cham**, the results are quite acceptable but studies need to be extended to deal with non-homogeneous lines. For the huge collection of **Textline-Manuscript-Cham**, with many different cases, an adaptation of the proposed method for inscriptions combined with available deep learning models proves the relevance of our approach.

Since the results from the segmentation task may not completely correct all the parts of the text line it still be recognized by the recognition model. So one of the aspects that can be used to evaluate the performance can be based on the recognition task.

CHAPTER 6

Text line transliteration

In this chapter, we present the text line transliteration problem on the Cham manuscripts documents. Unlike standard character recognition or machine transliteration, the model has to both recognize then translate written text from Middle Cham script to Latin script. Taking into account the advantage of each sequence-to-sequence model and transformer-based model, a two-step strategy has been proposed for further improving the transliteration performance.

6.1 Introduction

6.1.1 Definition

Transliteration involves the process of converting, character by character (also following the transliteration rules), a text from one source script to another script (for instance α in Greek to 'a' in Latin). This process can be applied with input text (text-to-text) or input image (image text-to-text) depending on the input data. In this work, we have images of documents as input, the system needs to recognize the text in the input image then correctly transcribe it to the target

writing system. Transliteration can be viewed as an image-to-text task, just like an Optical Character Recognition (OCR) task. So, we studied different approaches from standard OCR for our problem. This task can be done at different levels: character, glyph, word, or sentence level. Specifically, we focus on the methods using text lines as input. More precisely, we have as input the image of each line of text (written in Middle Cham), and the output contains the corresponding text file in the target script (Latin).

6.1.2 Challenges

Due to the complex transliteration rules of Cham script, the transliteration of Cham documents poses several challenges:

- **Writing script**: As the original writing Cham language is derived from Indian Brahmi, by default, each symbol transcribes an open syllable consisting of a consonant followed by its inherent vowel a; other vowels are transcribed using graphic symbols added above, beside, or below the graphic symbol of the consonant (See Fig. 6.1). The sophisticated position of these vowels can mislead the recognition model.

- **Transliteration rules** Specific characteristic of the writing style of Cham language: there is no space between words. Model has to learn how to separate words. In our work, the ground truth with separated words has been provided by our expert (See Fig. 6.2).

- **Similar characters**: The alphabet system contains many similar characters. In addition, the writing style of the different writers can lead to confusion during the recognition of the characters and the transliteration process. (See Fig. 6.3)

- **Less available resource**: Cham language can be considered as an under-resource language. As there is no existing dataset for Cham document recognition and transliteration, we have to annotate our dataset. Preparing ground truth is a time-consuming and tedious task. Therefore, the amount of data available for training and testing is quite small. This leads to the problem of converging in training recognition models, especially deep learning based models.

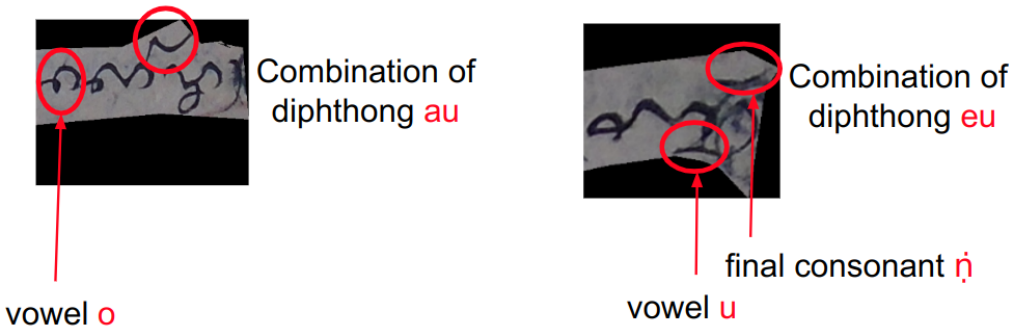


Figure 6.1: The formation of diphthong from vowel in Cham language

Figure 6.2: Illustration of separation word in the text line. Red straight lines indicate the space between words.

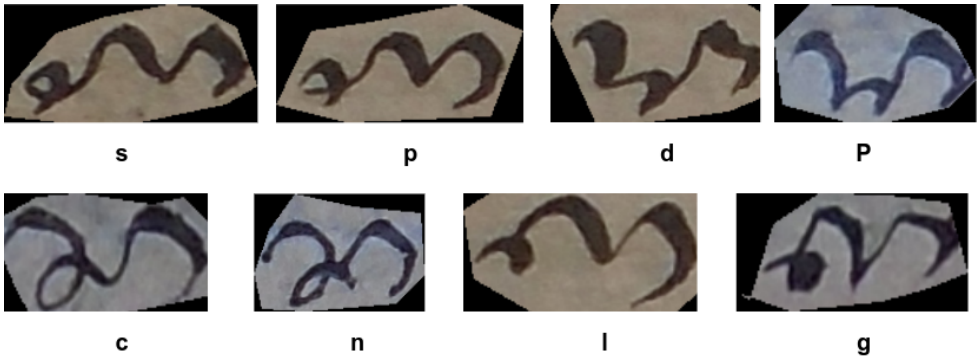


Figure 6.3: Example of similar characters in Cham script

6.2 Two-step Sequence Transformer Approach (TSST)

As mentioned in section 3.3, in the Cham language, the transliteration of certain characters depends not only on the visual characteristics but also on the meaning of the word or the context of the sentence. Moreover, some characters are very similar as shown in the figure 6.3. We can observe that the character s is similar to the character p. The same goes for the characters d and P, c and n or l and g. Therefore, using a one-step approach as commonly proposed in typical recognition processes may not handle this case. In our work, we have evaluated the performance of the SOTA one-step methods such as Shi et al. (2015), Kang et al. (2022) and Du et al. (2022). Experimental results (presented in 6.4.1 section) show that their models make the mistake with close characters. From these observations, we propose a two-step approach for Cham transliteration. Firstly, we propose a scenario for recognition where similar characters are considered as unique characters, then we use the transformer model which considers both visual and context information to adjust the prediction when it concerns similar characters to be able to distinguish them. To serve with this above idea, for the training process, two ground truths have been created. y^m corresponds to the ground truth where the annotation of similar characters is the same. For instance, the annotation of the characters s and p is set to s. This ground truth is used during the first step. In the second step, the ground truth y, the annotation of each character is set according to their real meaning. In other words, the annotation of the character s is set to s, and the one of p is set to p. This ground truth is used during the second step. The list of similar characters and their annotations used during the first step is shown in table 6.1. Let's note that the characters p and P correspond to two different sounds in Cham

Table 6.1: Ground truth of similar characters

Similar characters	ground truth y^m
s, p	s
d, P	d
c, n	n
l, g	l
nd, N	N

language (likewise n and N). Figure 6.4 illustrates the architecture of the proposed method for the text line transliteration of the Cham document. This architecture consists of an attention

encoder-decoder module and a multi-modal transformer module. The attention encoder-decoder module follows the architecture proposed in Kang et al. (2018). The Encoder module consists of several convolution layers to extract the immediate features from the input image and recurrent layers to learn mutual information and the sequential nature of text existing in the image Kang et al. (2018). The second module is a multi-modal transformer. It consists of a stack of some transformer decoder layers Vaswani et al. (2017). The processing through each module of the proposed architecture is detailed in the following paragraphs.

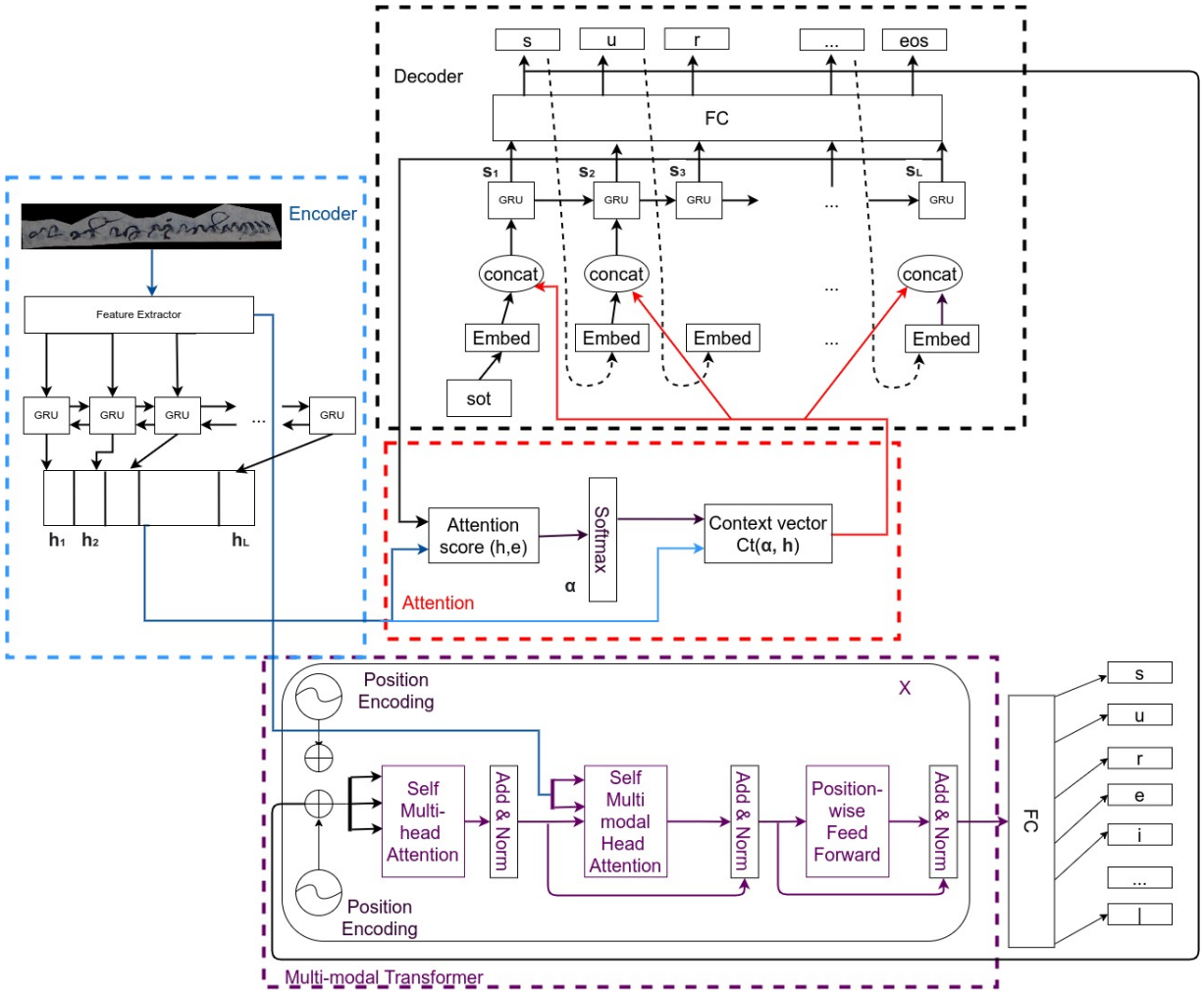


Figure 6.4: Proposed method TSST for text transliteration for images of Cham manuscripts

6.2.1 Attention Encoder-Decoder

The first module is the attention encoder-decoder (see Figure 6.4, the encoder is the blue block, the attention module is the red block and the decoder is the black block). The main objective of this module is to extract features from the input image (encoder network) and then decode these extracted features into their corresponding text. For more accurate decoding, an attention mechanism was used to focus on the appropriate encoder features.

Encoder

As the input of the system is an image, the use of a CNNs, as a feature extractor, is relevant to extract high-level features. Let's note $I \in R_{H \times W \times C}$ represents the input image in the feature space where H, W, C are respectively the height, width, and number of channels of the image ($C = 3$ for RGB image, $C = 1$ for gray-scale or binary image). Through the feature extractor network, we obtain the intermediate feature $V_f \in R_{H_v \times W_v \times E}$ where E is the number of channels of the feature map. The V_f is flattened in the space $R_{L \times E}$ where L is equal to $H_v \times W_v$. V_f will be the input of the RNN which returns the hidden state h_i at every time step $(i = 1, 2, ..., L)$.

There are many CNNs such as VGG-19 Simonyan and Zisserman (2014), Resnet He et al. (2015), EfficientNet Tan and Le (2019) that can be chosen as a backbone. However, their parameters such as kernel size, and stride of some layers need to be adjusted to our dataset (the maximum number of characters in the text line image). The details about the customization feature extractor network are presented in the section 6.3.2 and the comparison with other networks is given in the section 6.4.1.

Attention

Attention was presented in the work of Bahdanau et al. (2014) and Luong et al. (2015b) to deal with the inductive bias and representation bias of RNN encoder-decoder. More specifically, at every time step of the decoding phase, instead of reading equally extracted features V_f from the encoder network, a different weight α_{ij} will be computed based on the relevance of its features with the current decoding step. The context output vector c_t which represents for current input feature for the decoding at this step will be calculated based on the weighted sum of hidden

states features. We used the same attention scheme which is proposed in Bahdanau et al. (2014) and used in Kang et al. (2018). A summary of the computation of attention is given below in Algorithm 7.

Algorithm 7 Computation of attention at the step t

Require: List of hidden states from the encoder h_1, h_2, ..., h_L; embedded state from decoder e_t.
Attention score: score(h_k, e_t), k = 1, ... , L {score here is computed by feed-forward neural network.}
Attention weights:

$$\alpha_k^t = \frac{e^{(score(h_k, e_t))}}{\sum_{i=1}^{L} e^{(score(h_i, e_t))}} \tag{6.1}$$

context vector:

$$c_t = \sum_{k=1}^{L} \alpha_k^t h_k = \alpha_1^t h_1 + \alpha_2^t h_2 + \alpha_3^t h_3 + ... + \alpha_L^t h_L \tag{6.2}$$

Decoder

During this step, the prediction of each character is carried out by decoding the encoding feature with the previous prediction of characters obtained during the first step. This process can be formulated as:

$$p(y_t|y_1, ...y_{t-1}) = g(y_{t-1}, h_{t-1}, c_t) \tag{6.3}$$

where y_t is the current character we want to predict, y_{t-1} is the previous prediction, h_t is the hidden state and c_t is the context vector both at the step t. In other words, the decoding process starts by considering the special character "start of sentence" (SOT) as the first input character. The decoding process ends when the "end of sentence" prediction (EOS) occurs or reaches the maximum time step (maximum number of characters in one line of the data set). In detail, the SOT character is firstly converted into the embedding space of characters with the following equations.

$$embed = Embedding(y_{SOT}) \tag{6.4}$$

where y_{SOT} represents for start-of-sentence and the $Embedding$ function allows converting the character into its representation $embed$ in the space vector. This $embed$ representation is con-

catenated with the context vector c_1 as the input for decoding the first character. The first output y_1 is obtained as follows :

$$y_1 = \omega([embed, c_1]) \tag{6.5}$$

where ω is the trainable parameters representing RNN layers followed by fully connected layer. For each iteration t, this process is repeated by using the prediction of the previous character at $t-1$ and the context vector c_t as the input for decoding at iteration t.

$$embed_t = Embedding(y_{t-1}) \tag{6.6}$$

$$y_t = \omega([embed_t, c_t]) \tag{6.7}$$

6.2.2 Multi-modal Transformer

This module proposes to transform and adjust the prediction from the decoder phase to the real ground truth by combining the context of the sentence and the visual features. (violet block in the figure 6.4).

In the literature, the RNN encoder-decoder model has become a State-of-the-art approach for sequence modeling problems such as machine translation, sentences recognition Bahdanau et al. (2014), Cho et al. (2014). However, the problem with the encoder-decoder model is the process at each time step. Both the encoder and the decoder have to wait for the finish of $t-1$ steps to process the t step. This phenomenon will limit the parallelization capacity in training, especially for the longer sequence lengths, it is very time-consuming and computationally inefficient. To prevent the sequential RNN model, there are some efforts to Kalchbrenner et al. (2016), Meng et al. (2015) only use CNNs block to adapt with the sequence modeling. However, the results of CNNs-based could not outperform that of RNN on standard benchmark datasets Gehring et al. (2017). Therefore, Transformer Vaswani et al. (2017) is proposed to exclude recurrent and CNNs by using only the self-attention mechanism to represent the global dependencies between input and output sequence.

The overview of the transformer model is illustrated in the figure 6.5. It consists of two modules: Encoder module (left) and Decoder module (right). The core block of each module is the self-

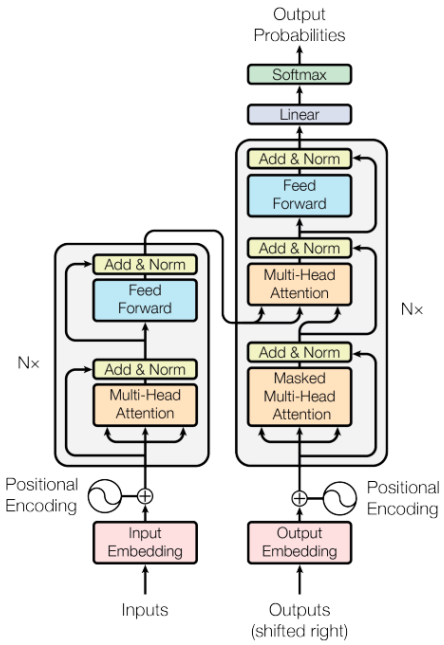


Figure 6.5: Transformer architecture proposed by Vaswani et al. (2017)

attention layer which is the combination of the Multi-Head Attention and the position-wise fully connected feed-forward network.

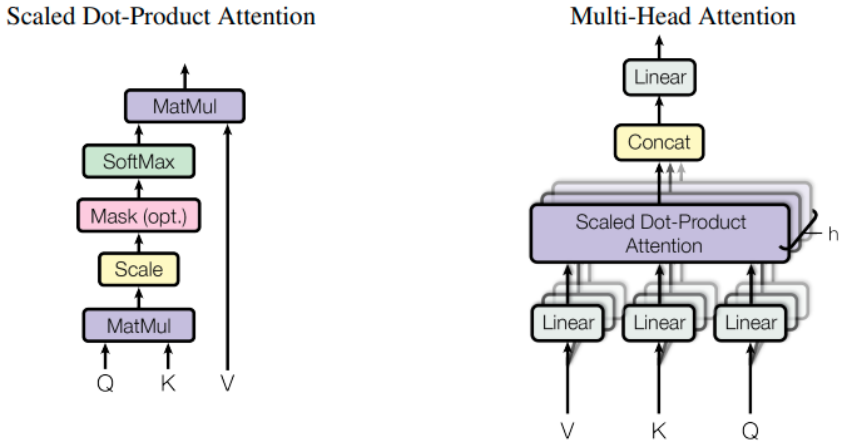


Figure 6.6: Scaled Dot-Product Attention and Multi-Head Attention Vaswani et al. (2017)

Multi-Head Attention

Multi-Head Attention is the multi-head self-attention which is done by repeatedly multiple Scaled Dot-Product Attention (see Fig. 6.6). Let's assumption the output of token i from em-

bedding is $x_i \in R^{d_{emb}}$ where d_{emb} is the output dimension of the embedding network.

Scaled Dot-Product Attention is computed as :

$$head_1(Q_1, K_1, V_1) = softmax(\frac{Q_1 K_1^T}{\sqrt{d_k}})V_1 \tag{6.8}$$

where: Q_1 = $x_i \times W_i^{Q_1}$, K_1 = $x_i \times W_i^{K_1}$, $V_1 = x_i \times W_i^{V_1}$; $\sqrt{d_k}$ is the scaling factor

$W_i^{Q_1} \in R^{d_{emb} \times d_q}$, $W_i^{K_1} \in R^{d_{emb} \times d_w}$, $W_i^{V_1} \in R^{d_{emb} \times d_v}$ are the trainable weights.

Multi-Head Attention is computed as :

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^0 \tag{6.9}$$

where $head_2, ...head_n$ are the repeat computation Scaled Doc-Product Attention of x_1 with different initial of three trainable weights $W_i^{Q_1}$, $W_i^{K_1}$, $W_i^{V_1}$; $W^0 \in R^{hd_v \times d_{emb}}$.

Position-wise fully connected feed-forward

Position-wise fully connected feed-forward is just simply two linear transformations with ReLU activation function.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{6.10}$$

where $x \in R^{d_{model}}$ is the input feature. $W_1 \in R^{d_{model} \times d_{ff}}$, $W_2 \in R^{d_{ff} \times d_{model}}$ are trainable weights.

Positional Encoding

The problem with the computation above is the missing order of the sequence, so positional encoding is used to inject the relative position between tokens in the sequence.

$$PE_(pos, 2i) = sin(pos/10000^{\frac{2i}{d_{model}}}) \tag{6.11}$$

$$PE_(pos, 2i + 1) = cos(pos/10000^{\frac{2i}{d_{model}}}) \tag{6.12}$$

where pos is the position of a token. Encoder and Decoder are constructed based on the self-attention layer above. Encoder is composed of a stack of identical self-attention layers, while Decoder has the same architecture but it has one more sub-layer (See Fig. 6.5) to perform multi-head attention with output from Encoder.

Multi-modal Transformer

We adopt the Transformer approach detailed above by using only Decoder blocks and then customize it by considering both the visual and text features. The fusion between the two kinds of features allows the model to obtain a better representation for the final prediction. Instead of a simple concatenation as in Tang et al. (2021), the two modalities are used interactively directly through the transformer model. The input of this module is the feature extractor V_f and the output S_f is obtained during the decoding phase. First, we compute Q_s from S_f by Scaled Dot-Product Attention as follows:

$$Q_s = (\frac{QK^T}{\sqrt{d_k}})V \tag{6.13}$$

where $Q \in R^{d_{model}}$, $K \in R^{d_{model}}$, $V \in R^{d_{model}}$ are calculated from S_f as below :

$$Q = W_Q \times S_f; K = W_K \times S_f; V = W_V \times S_f$$

where: W_Q, W_K, W_V are the trainable weights. Q_s is then used in the multi-modal transformer whose output S_sv is computed as follows :

$$S_sv = (\frac{Q_sK_v^T}{\sqrt{d_k}})V_v \tag{6.14}$$

where: the value V_v and the key K_v are computed from the visual feature V_f (output from feature extractor network) as below :

$$V_v = W_{V_f} \times V_f; K_v = W_{K_f} \times V_f$$

where W_{V_f} and W_{K_f} are the trainable weights.

The output S_{sv} is the input of the linear layer to obtain the final prediction.

6.2.3 Training Objective

Our model is trained with a Cross Entropy Loss which compares the differences between prediction and ground truth. Due to the small amount of training data which can lead to an over-fitting, we use Label Smoothing regularization Szegedy et al. (2016) to make the model have higher entropy when predicting. The final objective function is the combination of two losses:

$$L = \frac{1}{M}\sum_{i=1}^{M}(L_1(\hat{y}_1^i, y_i^m) + L_2(\hat{y}_2^i, y_i)) \tag{6.15}$$

where $L()$ is the Cross Entropy Loss, $\hat{y}_1^i$, $\hat{y}_2^i$ are the prediction of the model after the decoding step and after applying the multi-modal transformer on image i, y_i^m, y_i are respectively the ground truth with annotation for similar characters and the one for real characters in image i. M is the number of samples in the training set.

6.3 Training setting and evaluation metrics

6.3.1 Data preparation

To evaluate the proposed method, we employ the **Transliteration-Manuscript-Cham** dataset which consists of 4507 annotated text line images. We split the dataset into three sets of 3717/401/489 text line images for training, validation, and testing, respectively. In total, the dataset contains 70 characters (characters and some special notations). Figure 6.7 and 6.8 show respectively the distribution of the number of characters in the text lines and their width. We can note that the range of the number of characters in text lines varies from a few letters up to 99 letters. Similar to the distribution of the number of characters, the distribution of width is quite large from 100 pixels to 2000 pixels.

6.3.2 Parameters Settings

For the training, we resize the height of all images to 100 pixels while keeping the width-to-height ratio. If the new width is over 1000 pixels, it is still reshaped to 1000 pixels without keeping the width-to-height ratio. Based on the resized height (100 pixels) and limited width

of the image, we adjust some layers of the feature extractor network (in the encoder network) so that the output feature has the time step T around the maximum number of characters in one text-line. We can select another value for the height then we can customize the feature extractor so that the output feature has the time step T around the maximum number of characters in the one text-line. With a height 100 pixels and a maximum width is 1000 pixels, the max value of T is 125 (the maximum number of characters in one text-line is 99). The customization applied on the VGG-19 is detailed in table 6.2 with a stride size and a kernel size for each specific layer. We found that using the Average Pooling layer got a slight improvement compared with the default Max Pooling layer.

For the EfficientNet-based, Resnet-based, there are many variations of these pre-trained networks. So, in this experiment, we just take one representation of each network. For the

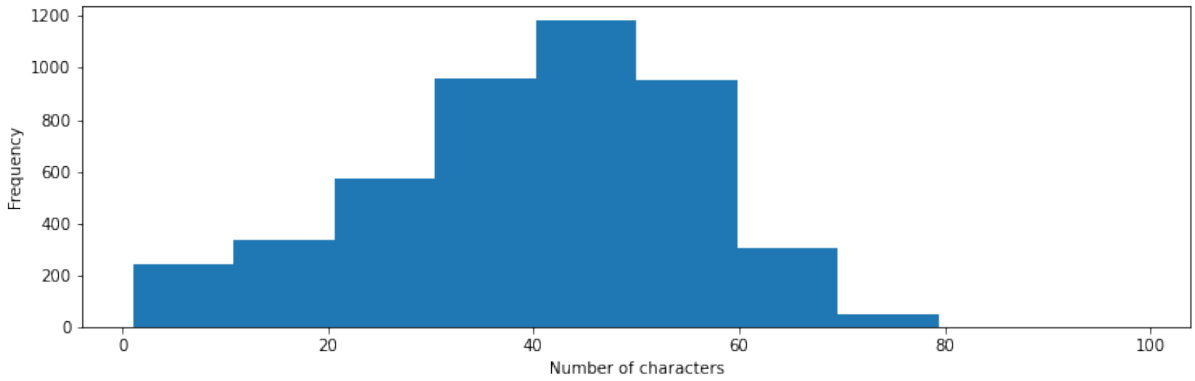


Figure 6.7: Distribution of the number of characters in text lines.

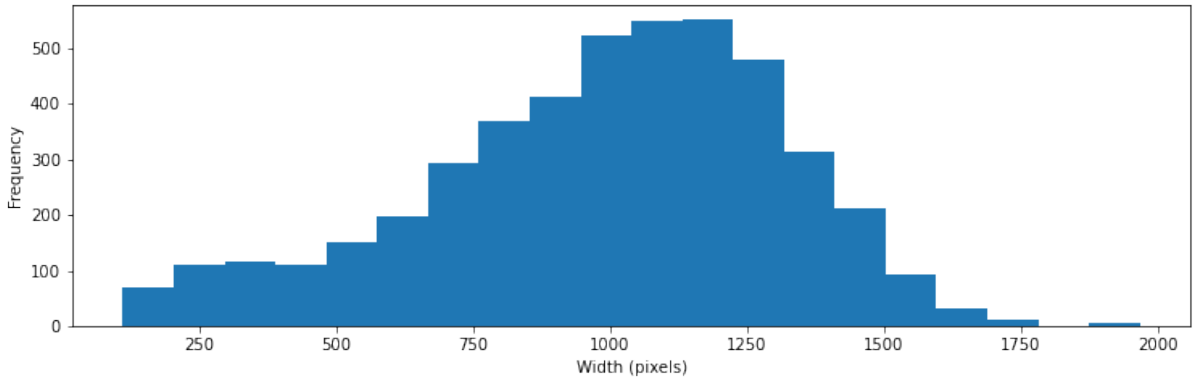


Figure 6.8: Distribution of the width of text lines.

Table 6.2: Customization of VGG-19

Layer name	Kernel size	Stride size
Average Pooling 1	4x2	4x2
Average Pooling 2	2x2	2x2
Average Pooling 3	2x2	2x2
Average Pooling 4	2x1	2x1
Average Pooling 5	2x1	2x1

EfficientNet-based Tan and Le (2019), we used the pre-trained EfficientNet-B0 with additional unit Squeeze-and-Excitation (SE) block Hu et al. (2018). For the Resnet-based, we used a modification of Resnet-50 proposed in Cheng et al. (2017) with the pre-trained model from Baek et al. (2019).

Inspired by the masked language model training from Devlin et al. (2018), we use this strategy by randomly replacing 5% of the characters during the training phase with a "new" character **MASK**. The details parameter of the transformer model is summarized in the table 6.3.

Table 6.3: Details of transformer model

Parameter	Value
dimension embedding (d_{emb})	256
dimension model (d_{model})	256
number of head (h)	8
Dimension of the feadforward (d_{ff})	2048

To prevent overfitting and to increase the number of samples of some types of degradation existing in the documents, different strategies of data augmentations have been used. We randomly applied transformations such as blur, dropout, solarize, affine transform, perspective transform, and crop. All these transformations are applied with a probability of 0.3. This value, fixed experimentally, is a good trade-off making it possible to ensure the convergence of the model.

6.3.3 Metrics

In the literature, Character Error Rate (CER) and Word Error Rate (WER) are often used to evaluate the performance of text recognition tasks. In our work, the transliteration task needs to recognize the Cham characters before proposing their corresponding "sound" in Latin script.

So, we first decide to use these two metrics to evaluate the performance of the methods.

- Character Error Rate (CER) : measures the performance of the model lying the Levenshtein distance between the ground truth and the prediction output. This metric is computed as:

$$CER = \frac{S + D + I}{S + D + C} \tag{6.16}$$

  where:

  - S : is the number of substitutions **characters**
  - D : is the number of deletions **characters**
  - I : is the number of insertions **characters**
  - C : is the number of correct **characters**

- Word Error Rate (WER) : is computed like CER, but it compares the results at the word level instead of the character level as in CER.

$$WER = \frac{S + D + I}{S + D + C} \tag{6.17}$$

  where:

  - S : is the number of substitutions **words**
  - D : is the number of deletions **words**
  - I : is the number of insertions **words**
  - C : is the number of correct **words**

For both metrics, the lower results highlight the better method. In another aspect, the transliteration in our work can be also considered as a machine transliteration task which compares the translated text from one language to another language. So, we adapt Bilingual Evaluation Understudy (BLUE) score Papineni et al. (2002) which is usually used to validate performance on this task.

- BLUE score defined as :

$$BLUE = BP.exp(\sum_{n=1}^{N} w_n log p_n) \tag{6.18}$$

where BP is the brevity penalty factor, which is a correction factor that penalizes shorter translations; w_n is the weight of each n-gram score, and p_n is the precision score for each n-gram. Unlike the standard machine translation task, ground truth may contain more than one, here we have only one predicted text and ground truth text so, p_n can mathematically as :

$$p_n(\hat{y}, y) = \frac{\sum_{s \in G_n(\hat{y})} min(C(s, \hat{y}), C(s, y))}{\sum_{s \in G_n(\hat{y})} C(s, y)} \tag{6.19}$$

where $\hat{y}$, y are the prediction and ground truth of the text line, respectively; and

$$\sum_{s \in G_n(\hat{y})} C(s, y) = \text{number of n-substrings in } \hat{y} \text{ that appear in y} \tag{6.20}$$

For this metric, higher results are better methods.

6.4 Experimental results

While the proposed method combines many components, to evaluate the role of different modules, an ablation study has been conducted. Moreover, the comparison of the proposed method with the State-of-the-art (SOTA) methods in the literature is also described.

6.4.1 Ablation study

In this section, we analyze the role of the backbone feature extractor, type of decoder, number of layers of multi-modal transformer, and other configurations of the training model in transliteration performance.

Evaluation of different backbone feature extractors

The objective of the first experiment is to evaluate the performance of the encoder network. Encoder network includes two modules: feature extractor and two layers Bidirectional Gate Recurrent Units (BGRU). For a fair assessment, we keep the same setup of two layers BGRU while the Multi-modal Transformer is not used. Therefore, the model includes only the encoder network with the attention decoder module. We selected three kinds of typical CNNs-based feature extractor: VGG19-based Simonyan and Zisserman (2014), EfficientNet-based Tan and Le (2019), Resnet-based He et al. (2016). The results are presented in Table 6.4. As a first

Table 6.4: Ablation study of feature extractor model

Our model with different feature extractor model	WER (%)	CER(%)	BLUE-3 (3-grams)	Number of Parameters (Million)
Resnet-based Cheng et al. (2017)	8.46	3.41	0.85	11.0
EfficientNet-based Tan and Le (2019)	7.13	2.48	0.86	3.7
VGG19-based Simonyan and Zisserman (2014)	6.46	2.27	0.87	20.1

observation, we note that the backbone with VGG-19 obtains the best results on the WER and CER, BLUE metrics, VGG19-based improves WER results by 1.14% and 0.21% respectively for Resnet-based and EfficientNet-based. It is the same on BLUE with the slight improvement 0.02% and 0.01%. However, the number of parameters used with VGG19 is quite high in comparison with Resnet and EfficientNet. A trade-off between the accuracy and inference time should be considered during the implementation. Some quantitative results are shown in figure 6.9. We see that the general mistake from all the models comes from the unclear characters and close characters.

Evaluation of decoder network types

In the second ablation study, we performed different experiments to evaluate the role of the decoder models. Here, all models use the same feature extractor network: our customization

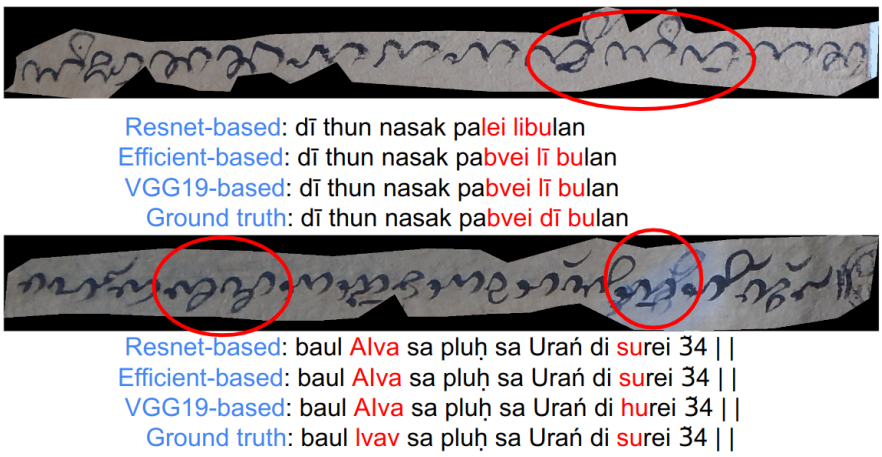


Figure 6.9: Quantitative results with different backbones

of VGG-19 is detailed above. For a fair comparison, all the weights are initialized from the pre-trained weights of VGG-19 in Simonyan and Zisserman (2014). To evaluate the role of the

Table 6.5: Ablation study of decoder module

Our model with different decoder module	WER(%)	CER(%)	BLUE-3(3-grams)
GRU-based Shi et al. (2015)	7.21	2.24	0.86
Transformer-based Kang et al. (2022)	6.78	2.61	0.87
Attention GRU-based Kang et al. (2018)	6.46	2.27	0.87

decoder model, three kinds of decoder models have been tested GRU-based like in Shi et al. (2015), Attention GRU Kang et al. (2018), Transformer Kang et al. (2022). Results are shown in Tab. 6.5.

We can observe that the three decoder models give quite good results for both metrics, around 7 % for WER and 2% for CER and 0.87 for BLUE. Decoding with GRU-based achieved better results in comparison with the Transformer-based method on CER metric. In terms of meaning and context of predicted text line (BLUE score), the Transformer-based method and Attention GRU-based improve slightly the results compared to using GRU-based only. The main problem with these models comes from the separating words and the close letters (See Fig. 6.10). We also observed that the transformer-based method gives correct results in some cases where both Attention GRU-based and GRU-based make the mistakes. Transformer is potentially a good approach but more data will be needed to build a good model. Some illustrations in good transliterations are given in figure 6.11.

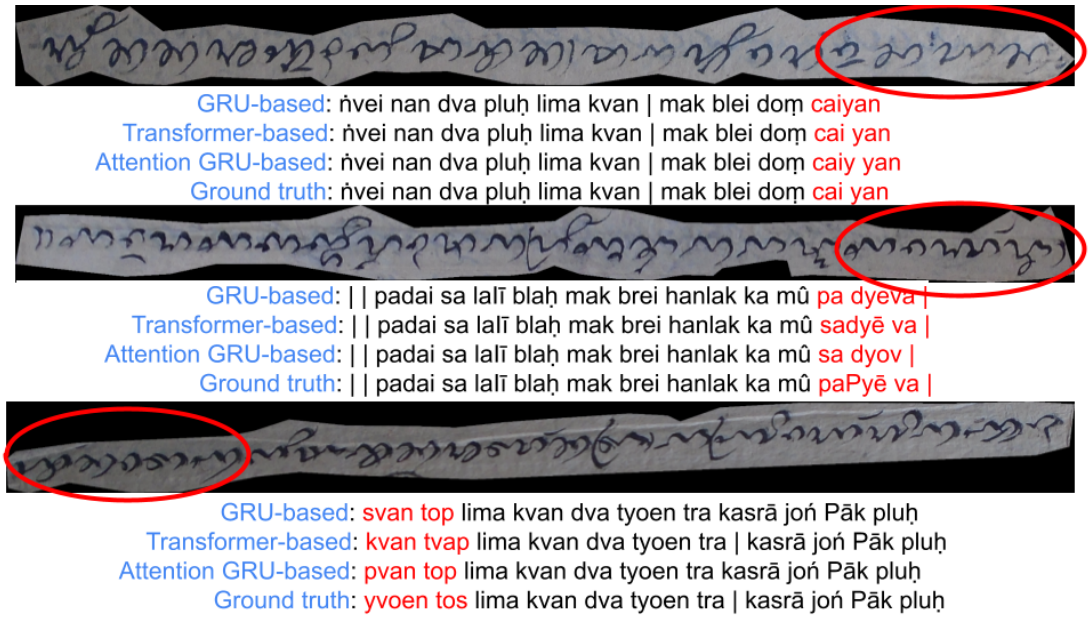


Figure 6.10: Qualitative results: Mistakes of the different decoders

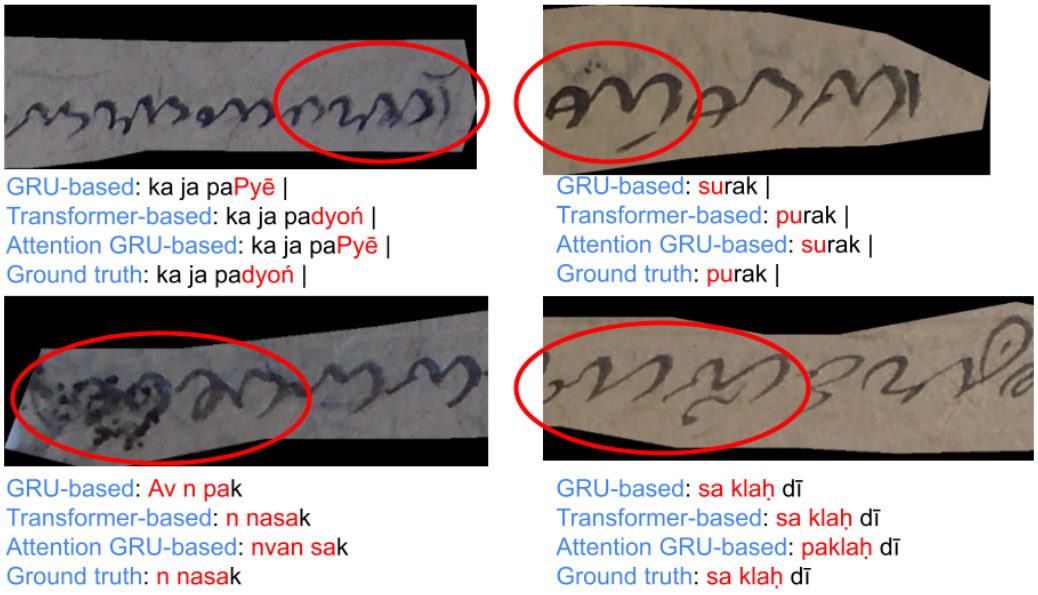


Figure 6.11: Qualitative results: Good cases with transformer model

Evaluation of the number of layers of multi-modal Transformer

In this experiment, we evaluate the change in configuration for the number of layers of Multi-modal Transformer. The results are given in the table 6.6. In general, the results are quite close

Table 6.6: Ablation study of the number of self-attention layer

Number of layer	WER (%)	CER(%)	BLUE-3(3-grams)
2	6.15	2.19	0.88
3	6.01	1.94	0.88
4	5.99	1.96	0.88
5	6.19	1.99	0.88
6	6.16	2.03	0.88

to each other. The best results are obtained at 3 or 4 layers. Using more layers does not improve the performance even making the results obtain lower.

Evaluation of other configurations

In this experiment, we summarize the results by considering the use of a two-step strategy. The results presented in the table 6.7. Experimental results confirm the robustness of the strategy of

Table 6.7: Ablation study of two-step strategy and using pre-trained model

Setting	WER (%)	CER(%)	BLUE-3 (3-grams)
Customization of VGG-19 + Attention-GRU decoder	6.46	2.27	0.87
TSST (w/o recognition of similar characters)	6.42	2.17	0.88
TSST (Customization of VGG-19 + Two-step)	5.99	1.96	0.88

the two-step approach which increases on WER, CER, and BLUE score. Thanks to the similar character recognition step, our model improves significantly for WER and CER, which respectively reach 5.99% and 1.96% in comparison without using this step. Figure 6.12 illustrates the improvement of the proposed method by comparing our approach Kang et al. (2018) without the Multi-modal transformer. The results show that the method can provide more correct predictions in the case of similar characters such as (s,p) or (d,P). In other cases of similar characters (l,g), (c,n) we did not see a big improvement in these cases. This can be explained by the fact, that (c,n) and (l,g) are still observed by the differences inside of character while for (nd, N) we do not have too many samples to train efficiently.

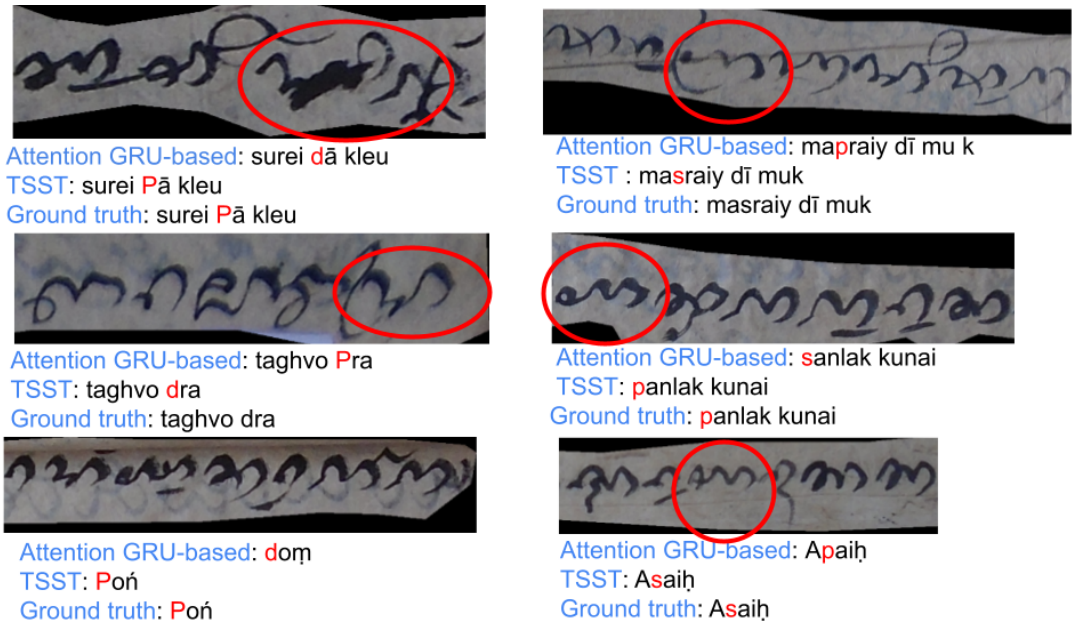


Figure 6.12: Qualitative results: Comparison of our approach with the baseline attention decoder method

6.4.2 Comparison with the state-of-the-art approaches

We compare the performance of the proposed method with that of different approaches in the literature. The overview of architecture and training process of each model is presented in the table 6.8. The summary results are given in the table 6.9. The proposed method outperforms all the studied methods of the literature. We found that models such as Atienza (2021) and Borisyuk et al. (2018), which only use CNNs or FC (fully connected) layer are not adapted to our data. They can not learn properly the implicit rules, this is why the results are not good. This can be explained by the fact that these kinds of models have been proposed for short text lines while our dataset contains both short and long ones. So the models are not well-trained. The iterative approaches like Na et al. (2022), Fang et al. (2021) are better than the two previous methods which do not use RNN, but the errors are quite high. The results with Li et al. (2021), Du et al. (2022) are promising but due to the small amount of data, the models tend to provide an over-fitting on the training set, hence the performances are still limited. The last architecture proposed by Shi et al. (2015) uses our customization feature extractor with the GRU layers and is quite appropriate for our volume of data. It shows its superiority in comparison with other approaches described above.

Table 6.8: Architecture and training of different approaches

Name	Feature Extractor	Decoder	Training Condition
CRNN Shi et al. (2015)	Our customization VGG-19	BGRU	Pre-trained on feature extractor Scratch for the Decoder
Rosetta Borisyuk et al. (2018)	Resnet-18	None	Scratch
SVTN Du et al. (2022)	CNN	None	Scratch
TrOCR Li et al. (2021)	Visual Transformer Dosovitskiy et al. (2020)	RoBERTa Liu et al. (2019)	Fine-tuning model from Pre-trained on the MJSynth and Synth-Text dataset
MATRN Na et al. (2022)	Resnet-45 + Transformer	Transformer + iterative decoding	Scratch
ABINET Fang et al. (2021)	Resnet-45 + Transformer	Transformer + iterative decoding	Scratch
ViTSTR Atienza (2021)	Visual TransformerDosovitskiy et al. (2020)	Non	Fine tuning of pre-trained model on the MJSynth and Synth-Text dataset

Table 6.9: Comparison with the state of the art methods

Model	WER	CER	BLUE-3(3-grams)
CRNN Shi et al. (2015)	7.41	2.24	0.86
Rosetta Borisyuk et al. (2018)	90.68	62.59	0.02
SVTN Du et al. (2022)	32.26	14.74	0.05
TrOCR Li et al. (2021)	16.47	8.76	0.74
MATRN Na et al. (2022)	54.02	38.11	0.30
ABINET Fang et al. (2021)	45.14	24.77	0.37
ViTSTR Atienza (2021)	94.91	72.62	0.03
TSST	**5.99**	**1.96**	**0.88**

6.4.3 Analysis of the wrong predictions

In this section, we analyze the incorrect predictions obtained with our model.

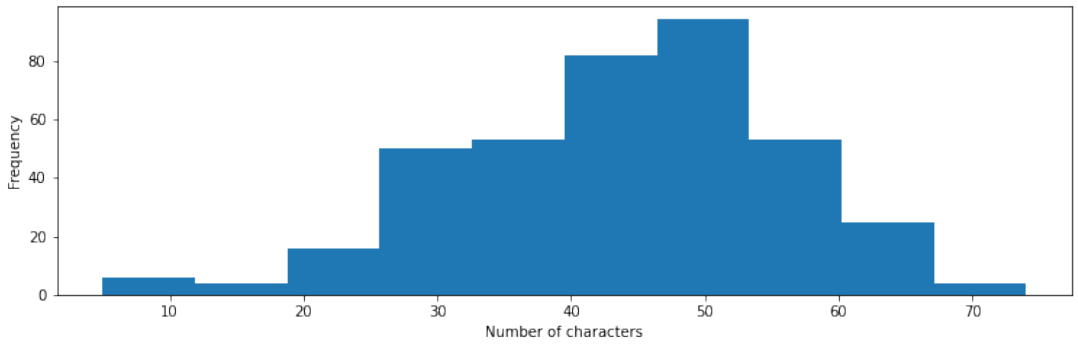


Figure 6.13: Distribution of the number of wrong predictions according to the number of characters by text lines

The first factor we want to evaluate is the number of characters in the text line. Figure 6.13 shows the distribution of the number of wrong predictions according to the number of characters by text lines. This figure has a distribution similar to the one of the training dataset, so the number of characters in the text line does not have a huge effect on the model. The second factor we want to analyze is the width of the text line image. Figure 6.14 presents the number of wrong predictiona according to the width of the text lines (after resizing the height to 100 pixels). This figure shows that most wrong predictions are obtained for text lines with a resized width

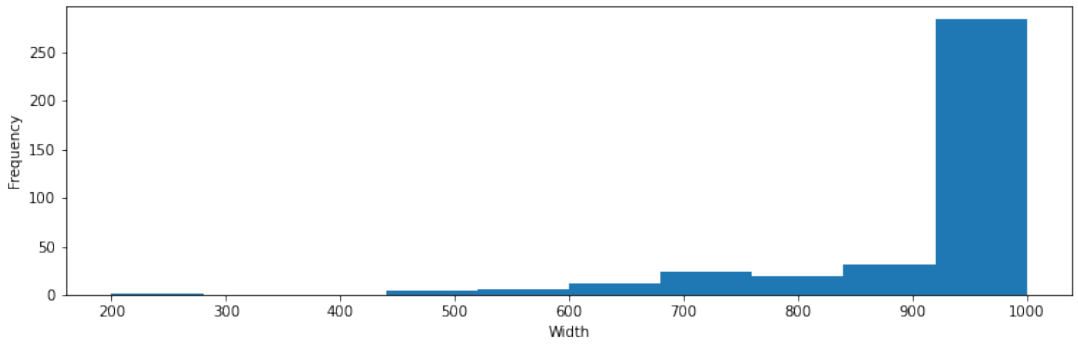


Figure 6.14: Distribution of wrong predictions according to the width of the text lines

at 1000 pixels. We assume that the modification of the width during the resizing distorts the image, leading to confusion of the model during the recognition. Figure 6.15 presents the most

wrong cases, the model misses some characters or finds non-existing characters. For the other

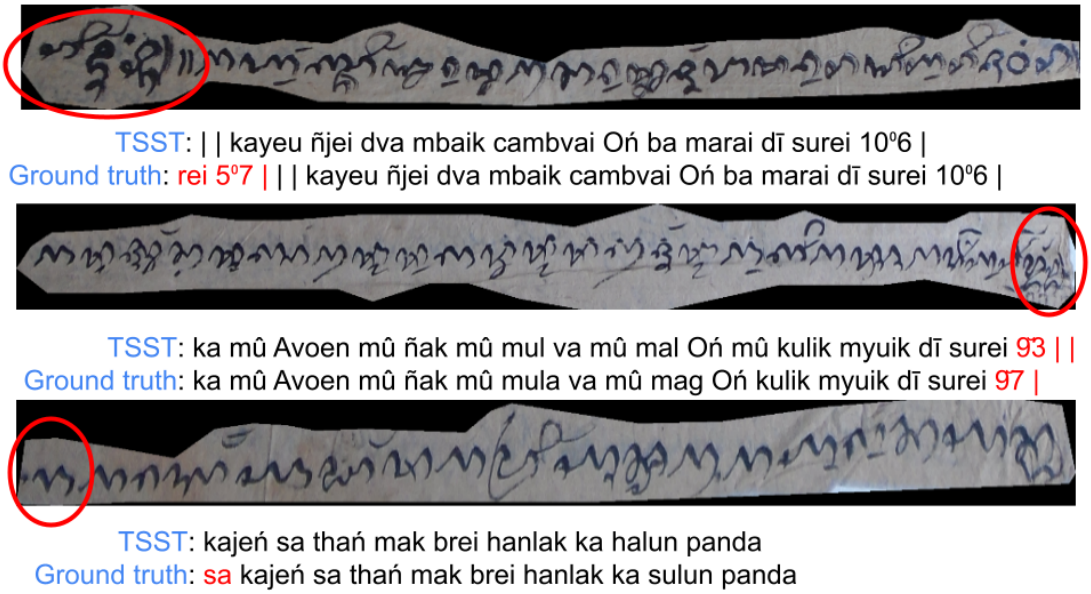


Figure 6.15: Examples of incorrect predictions for text lines with a large width

wrong predictions corresponding to text lines with large widths, the mistakes are simply due to uncertain characters or a bad separation between words. (See Fig. 6.16). Moreover, in the case

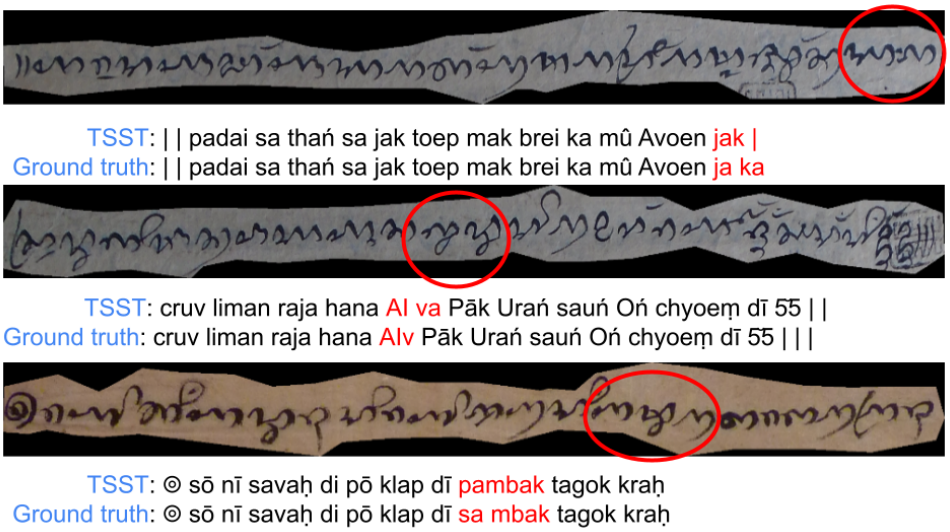


Figure 6.16: Examples of incorrect predictions for text lines with a large width in case of merging words.

of an image keeping the height-to-width ratio (resizing width not exceeding 1000 pixels), the

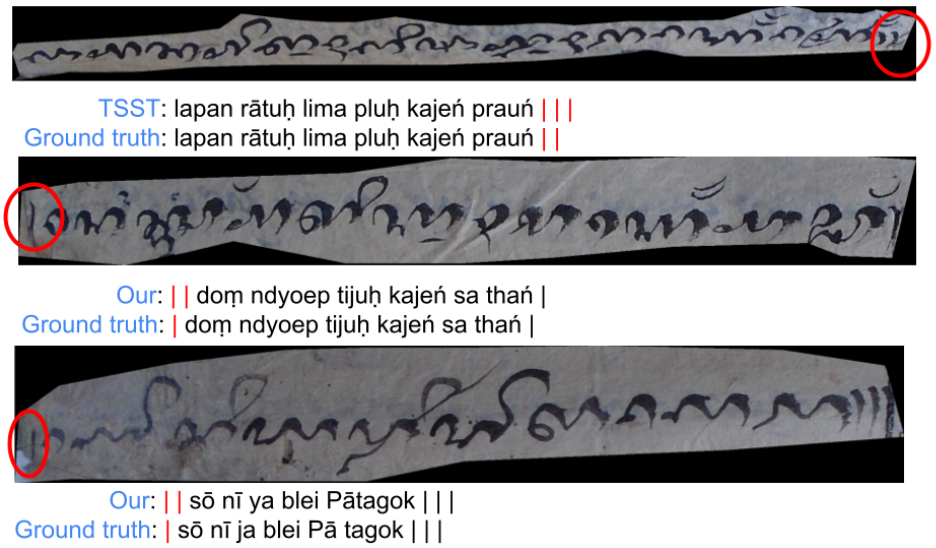


Figure 6.17: Examples of incorrect predictions in case of over-compliance rules with 'l'. There are three rules with 'l', one 'l' corresponds to the end of a single sentence, two 'l' corresponds to the end of a paragraph, and 'lll' means the end of a text.

common errors encountered are the following: over compliance rules with 'l') (See Fig. 6.17), merging words, noisy letters (can come from stamp or ink bleed through, noisy background (See Fig. 6.18) and the compliance rules with vowel "a" (See Fig. 6.19).

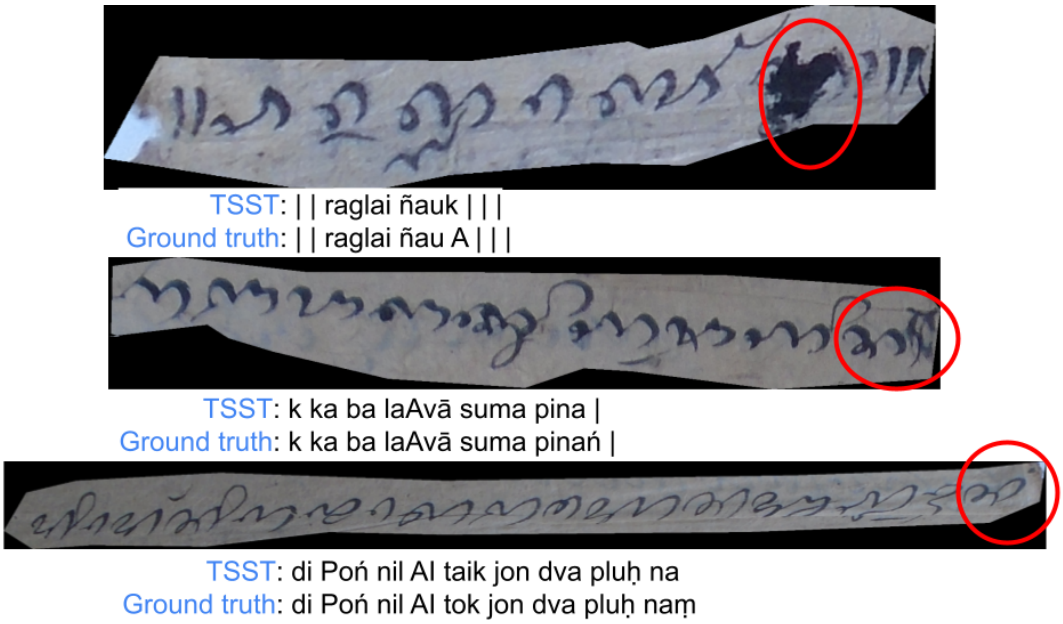


Figure 6.18: Examples of incorrect predictions in case of noisy image

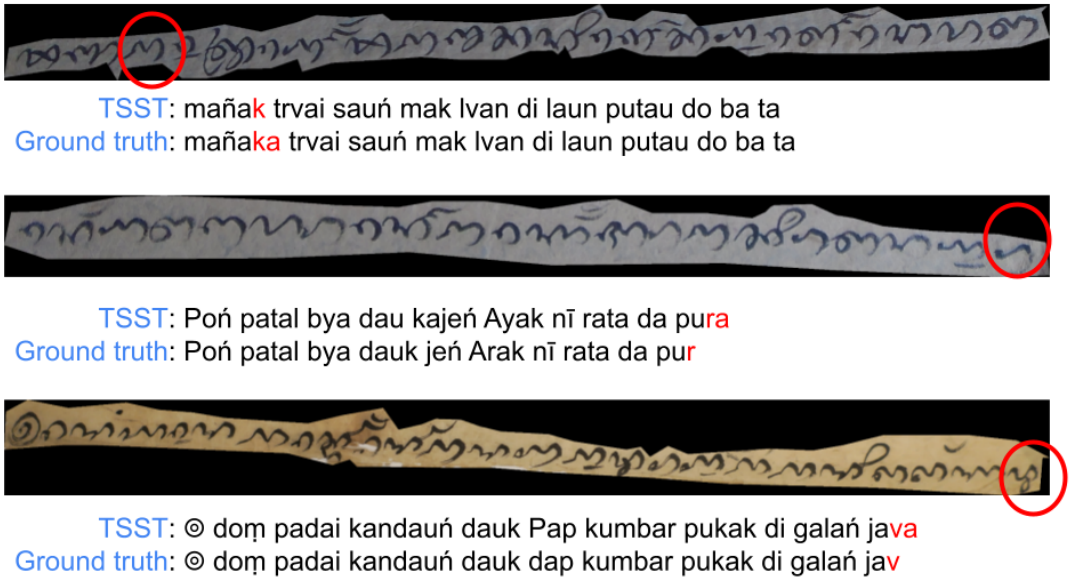


Figure 6.19: Examples of incorrect predictions in case of transliteration of vowel a. When the consonants end with a short stroke, it should not add the vowel a and vice versa.

6.5 Conclusions

In this chapter, we present the text line transliteration problem for the Cham language. Unlike the standard Optical Character Recognition (OCR) or transliteration (in the case of text-to-text), the system needs to do both tasks: recognition and then simultaneous transliteration. By considering the characteristics of the alphabet characters and the size of the text line image, we propose a two-step approach for the transliteration of Cham script to Latin script by combining a sequence-to-sequence model and a transformer model. By taking advantage of each model, we have significantly improved the results of transliteration in comparison with separate models on the **Transliteration-Manuscript-Cham** dataset. An extensive study proves the performance of the proposed method in comparison with other methods of the literature. From a deep analysis of the incorrect predictions, we have observed that the transliteration rules, the noisy documents, and the writing style are still open challenges for the community.

CHAPTER 7

Conclusions and Future works

7.1 Conclusions

The thesis has designed and developed different methods for analyzing and extracting information from Cham inscriptions and manuscript images. In summary, this dissertation has four main contributions:

- A deep learning method for image enhancement of Cham inscriptions images has been developed. In the proposed method, to improve the visual quality of document images, a global attention module has been added. This module allows to exploitation of the features at different levels By incorporating the training objective function, the whole model was trained in an end-to-end manner. The experiments on the **Denoising-Inscription-Cham** dataset have been conducted and have proved that the proposed approach improves the quality of the Cham inscription images in terms of noise removal and character readability.
- A simple and efficient method for text line segmentation based on the seam carving method was introduced. By considering the properties of the documents which contain

many diacritics, ascenders, and descenders, an appropriate set of cost functions has been designed to guide separation seams for greater accuracy between these components. Experiments on the **Textline-Inscription-Cham** and DIVA-HisDB dataset prove the relevance of our method. On the other hand, we also extend the proposed method for the manuscript documents by leveraging the pre-trained model. The extensive qualitative evaluation has proved the effectiveness of the proposed method compared with two off-the-shelf software.

- A two-step approach for the text line transliteration of Cham manuscripts by combining a sequence-to-sequence model and a transformer model has been developed. In the proposed method, first, a new scenario for recognition where similar characters are considered unique characters is added. Then, we use the transformer model which considers both visual and context information to adjust the prediction when it concerns similar characters to be able to distinguish them. an ablation study has been carried out to validate the contribution of each module of the proposed model. Moreover, the extensive experiments show that our method outperforms other approaches of the literature on our **Transliteration-Manuscript-Cham** dataset.

- The creation of three image datasets, named **Denoising-Inscription-Cham** for image enhancement, **Textline-Inscription-Cham** for text line segmentation and **Transliteration-Manuscript-Cham** for the transliteration. These datasets are built from two types of Cham documents: inscriptions and manuscripts. For each dataset, the characteristics of the Cham script bring new challenges to the research community.

7.2 Future works

Although the experiments carried out in this thesis, have shown that the results of the proposed methods for image enhancement, text line segmentation, and transliteration were quite promising. Many points can still be improved.

- **Unsupervised learning**: So far, most of the deep learning approaches used in our work are based on supervised learning which is data-hungry. However, data annotation is time-

consuming and tedious. The different scenarios for training deep learning models can be designed to take advantage of a huge amount of data available without annotation. Specifically, for the image enhancement problem, an initial experiment has been done with a weakly supervised approach. Although the results did not show a clear improvement, this idea is promising to solve some limitations imposed by traditional supervised learning methods. For the text line transliteration task, it is more difficult to directly apply unsupervised learning from scratch but using the modalities with a pre-trained model could offer potential improvement. Similarly, for the text line segmentation, the transfer learning proves the effectiveness on the manuscript images but on the inscription images, the results are quite limited so fine-tuning the pre-trained model with less ground truth could be considered.

- **Multi-modality, multi-task scenario** are also potential pathways. Firstly, with a multi-task scenario, we can explore the opportunity to build a unified model for different tasks such as image enhancement and text line transliteration, text line segmentation, and text line transliteration. In this way, the errors can be directly back-propagated between the tasks and, therefore, more suitable models can be defined. Multi-modality is another interesting direction to explore. For example, in the text line transliteration, we prove the effect of using two modalities (e.g. image and text) but more mutual interaction or feature enhancement between the modalities can be considered.

- **Paragraph Transliteration**: Right now, the transliteration for the Cham language is done at the text-lines level, but due to the boundary of the writing material, sometimes one word is written on two lines. Therefore, to correctly translate, context information should be added. This can be solved by using higher level transliteration such as paragraph level. The strategy for training multiple lines needs to be investigated in the future.

- **Human-Machine Interactive and End-to-End system**: Currently, models have been proposed for each task of the workflow. However, an end-to-end system capable of automatically performing document image analysis should be built by combining models from different tasks. To further enhance the outcome, a Graphical User Interface (GUI) allowing the user to interact with the system would be essential to validate the results

immediately and get user feedback.

Publications

This doctoral work has led to the following publications:

Conferences and workshops

- Tien Nam Nguyen, Jean-Christophe Burie, Thi Lan Le, and Anne-Valerie Schweyer : "On the use of attention in deep learning based denoising method for ancient Cham inscription images". Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I. Cham: Springer International Publishing, 2021.
- Tien Nam Nguyen, Jean-Christophe Burie, Thi Lan Le, and Anne-Valerie Schweyer: "An effective method for text line segmentation in historical document images". 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022.
- Tien Nam Nguyen, Jean-Christophe Burie, Thi Lan Le, and Anne-Valerie Schweyer: “Two-step sequence transformer based for Cham to Latin script transliteration”. International Workshop on Historical Document Imaging and Processing 2023

Bibliography

AbdulJaleel, N. and Larkey, L. (2003). "English to Arabic transliteration for information retrieval: A statistical approach". *Center for Intelligent Information Retrieval Computer Science, University of Massachusetts* (cited on p. 45).

Aberdam, A., Ganz, R., Mazor, S., and Litman, R. (2022). "Multimodal Semi-Supervised Learning for Text Recognition". *arXiv preprint arXiv:2205.03873* (cited on p. 42).

Ahn, J., Cho, S., and Kwak, S. (2019). "Weakly supervised learning of instance segmentation with inter-pixel relations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2209–2218 (cited on p. 83).

Ahn, J. and Kwak, S. (2018). "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4981–4990 (cited on p. 83).

Aiello, M., Monz, C., Todoran, L., Worring, M., et al. (2002). "Document understanding for a broad class of documents". *International Journal on Document Analysis and Recognition*, 5(1), pp. 1–16 (cited on pp. 22, 23).

Alberti, M., Vögtlin, L., Pondenkandath, V., Seuret, M., Ingold, R., and Liwicki, M. (2019). "Labeling, Cutting, Grouping: an Efficient Text Line Segmentation Method for Medieval Manuscripts". *CoRR*, abs/1906.11894 (cited on pp. 27–29, 92, 94–97, 102, 110, 111, 118).

Amin, A. and Fischer, S. (2000). "A document skew detection method using the Hough transform". *Pattern Analysis & Applications*, 3, pp. 243–253 (cited on p. 10).

Antonacopoulos, A., Clausner, C., Papadopoulos, C., and Pletschacher, S. (2013). "ICDAR 2013 competition on historical book recognition (HBR 2013)". In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE, pp. 1459–1463 (cited on p. 11).

Arbabi, M., Fischthal, S. M., Cheng, V. C., and Bart, E. (1994). "Algorithms for Arabic name transliteration". *IBM Journal of research and Development*, 38(2), pp. 183–194 (cited on p. 45).

Arvanitopoulos, N. and Süsstrunk, S. (2014). "Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts". In: *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 726–731 (cited on pp. 26, 27, 29, 92–96, 111, 112).

Asi, A., Saabni, R., and El-Sana, J. (2011). "Text Line Segmentation for Gray Scale Historical Document Images". In: *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. HIP '11. Beijing, China, USA: Association for Computing Machinery, 120–126 (cited on pp. 27, 29).

Atienza, R. (2021). "Vision transformer for fast and efficient scene text recognition". In: *International Conference on Document Analysis and Recognition*. Springer, pp. 319–334 (cited on pp. 43, 141, 142).

Avidan, S. and Shamir, A. (2007). "Seam carving for content-aware image resizing". In: *ACM SIGGRAPH 2007 papers*, 10–es (cited on p. 27).

Bae, J. H., Jung, K. C., Kim, J. W., and Kim, H. J. (1998). "Segmentation of touching characters using an MLP". *Pattern Recognition Letters*, 19(8), pp. 701–709 (cited on p. 25).

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., and Lee, H. (2019). "What is wrong with scene text recognition model comparisons? dataset and model analysis". In:

Proceedings of the IEEE/CVF international conference on computer vision, pp. 4715–4723 (cited on pp. 39, 42, 134).

Bahdanau, D., Cho, K., and Bengio, Y. (2014). "Neural machine translation by jointly learning to align and translate". *arXiv preprint arXiv:1409.0473* (cited on pp. 126–128).

Barakat, B. K., Droby, A., Alaasam, R., Madi, B., Rabaev, I., and El-Sana, J. (2021a). "Text line extraction using fully convolutional network and energy minimization". *CoRR*, abs/2101.07370 (cited on pp. 102, 118).

Barakat, B. K., Droby, A., Alasam, R., Madi, B., Rabaev, I., Shammes, R., and El-Sana, J. (2020). "Unsupervised text line segmentation". *CoRR*, abs/2003.08632 (cited on pp. 33, 34, 118).

Barakat, B. K., Droby, A., Kassis, M., and El-Sana, J. (2021b). "Text Line Segmentation for Challenging Handwritten Document Images Using Fully Convolutional Network". *CoRR*, abs/2101.08299 (cited on pp. 33, 34).

Barron, J. T. (2020). "A generalization of Otsu's method and minimum error thresholding". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, pp. 455–470 (cited on p. 13).

Bhunia, A. K., Chowdhury, P. N., Sain, A., and Song, Y.-Z. (2021). "Towards the unseen: Iterative text recognition by distilling from errors". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14950–14959 (cited on p. 43).

Bluche, T. (2016). "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition". *Advances in neural information processing systems*, 29 (cited on pp. 43, 44).

Bluche, T., Louradour, J., and Messina, R. (2017). "Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention". In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE, pp. 1050–1055 (cited on p. 43).

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). "Yolov4: Optimal speed and accuracy of object detection". *arXiv preprint arXiv:2004.10934* (cited on p. 75).

Bokser, M. (1992). "Omnidocument technologies". *Proceedings of the IEEE*, 80(7), pp. 1066–1078 (cited on pp. 36, 37).

Borisyuk, F., Gordo, A., and Sivakumar, V. (2018). "Rosetta: Large scale system for text detection and recognition in images". In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 71–79 (cited on pp. 39, 40, 141, 142).

Boulid, Y., Souhar, A., and Elkettani, M. Y. (2015). "Arabic handwritten text line extraction using connected component analysis from a multi agent perspective". In: *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 80–87 (cited on p. 26).

Brunelle, M. (2001). "Revisiting the expansion of the Chamic language family". In: *the International Conference of Champa, University of Hawai'i Press*, p. 26 (cited on p. 3).

– (2008). "Diglossia, bilingualism, and the revitalization of written Eastern Cham" (cited on p. 3).

Buades, A., Coll, B., and Morel, J.-M. (2005). "A non-local algorithm for image denoising". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 2. Ieee, pp. 60–65 (cited on pp. 17, 65, 78).

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). "Image denoising: Can plain neural networks compete with BM3D?" In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 2392–2399 (cited on pp. 20, 66).

Calvo-Zaragoza, J. and Gallego, A.-J. (2019). "A selectional auto-encoder approach for document image binarization". *Pattern Recognition*, 86, pp. 37–47 (cited on p. 104).

Casey, R. G. and Lecolinet, E. (1996). "A survey of methods and strategies in character segmentation". *IEEE transactions on pattern analysis and machine intelligence*, 18(7), pp. 690–706 (cited on p. 24).

Chamchong, R. and Fung, C. C. (2012). "Text line extraction using adaptive partial projection for palm leaf manuscripts from Thailand". In: *2012 International Conference on Frontiers in Handwriting Recognition*. IEEE, pp. 588–593 (cited on p. 26).

Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., and Zhou, S. (2017). "Focusing attention: Towards accurate text recognition in natural images". In: *Proceedings of the IEEE international conference on computer vision*, pp. 5076–5084 (cited on pp. 134, 137).

Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., and Zhou, S. (2018). "Aon: Towards arbitrarily-oriented text recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5571–5579 (cited on p. 40).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". *arXiv preprint arXiv:1406.1078* (cited on p. 128).

Choi, H. and Baraniuk, R. (1998). "Analysis of wavelet-domain Wiener filters". In: *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No. 98TH8380)*. IEEE, pp. 613–616 (cited on pp. 16, 18).

Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2011). "Convolutional neural network committees for handwritten character classification". In: *2011 International conference on document analysis and recognition*. IEEE, pp. 1135–1139 (cited on p. 39).

Cœdès, G. (1908). "Inventaire des inscriptions du Champa et du Cambodge". *Bulletin de l'École française d'Extrême-Orient*, 8(1/2), pp. 37–92 (cited on p. 3).

Cohen, R., Dinstein, I., El-Sana, J., and Kedem, K. (2014). "Using Scale-Space Anisotropic Smoothing for Text Line Extraction in Historical Documents". In: pp. 349–358 (cited on pp. 30, 32).

Coquenet, D., Chatelain, C., and Paquet, T. (2020). "Recurrence-free unconstrained handwritten text recognition using gated fully convolutional network". *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 19–24 (cited on p. 43).

Coquenet, D., Chatelain, C., and Paquet, T. (2022). "End-to-end handwritten paragraph text recognition using a vertical attention network". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cited on pp. 43, 44).

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). "Image denoising by sparse 3-D transform-domain collaborative filtering". *IEEE Transactions on image processing*, 16(8), pp. 2080–2095 (cited on pp. 20, 65, 78).

Deledalle, C.-A., Salmon, J., Dalalyan, A. S., et al. (2011). "Image denoising with patch based PCA: local versus global." In: *BMVC*. Vol. 81, pp. 425–455 (cited on pp. 18, 19).

Dengel, A., Bleisinger, R., Hoch, R., Fein, F., and Hones, F. (1992). "From paper to office document standard representation". *Computer*, 25(7), pp. 63–67 (cited on p. 23).

Dengel, A., Shafait, F., and Doermann, D. (2014). *Analysis of the Logical Layout of Documents* (cited on p. 22).

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805* (cited on p. 134).

Doermann, D., Tombre, K., et al. (2014). *Handbook of document image processing and recognition*. Vol. 1. Springer (cited on p. 22).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". *arXiv preprint arXiv:2010.11929* (cited on pp. 43, 66, 142).

Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., Du, Y., and Jiang, Y.-G. (2022). "SVTR: Scene Text Recognition with a Single Visual Model". *arXiv preprint arXiv:2205.00159* (cited on pp. 43, 124, 141, 142).

Dumpala, V., Kurupathi, S. R., Bukhari, S. S., and Dengel, A. (2019). "Removal of Historical Document Degradations using Conditional GANs." In: *ICPRAM*, pp. 145–154 (cited on p. 21).

Eglin, V. and Bres, S. (2004). “Analysis and interpretation of visual saliency for document functional labeling”. *Document Analysis and Recognition*, 7(1), pp. 28–43 (cited on p. 22).

Epigraphy, I. (1998). “a guide to the study of inscriptions in Sanskrit, Prakrit, and the other Indo-Aryan languages”. *South Asia Research Series. New York* (cited on p. 53).

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2017). “A comprehensive survey of mostly textual document segmentation algorithms since 2008”. *Pattern Recognition*, 64, pp. 1–14 (cited on p. 26).

Fang, S., Xie, H., Wang, Y., Mao, Z., and Zhang, Y. (2021). “Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7098–7107 (cited on pp. 42, 43, 141, 142).

Feldbach, M. and Tonnies, K. D. (2001). “Line detection and segmentation in historical church registers”. In: *Proceedings of sixth international conference on document analysis and recognition*. IEEE, pp. 743–747 (cited on p. 88).

Feng, M.-L. and Tan, Y.-P. (2004). “Contrast adaptive binarization of low quality document images”. *IEICE Electronics Express*, 1(16), pp. 501–506 (cited on p. 14).

Févotte, C. and Idier, J. (2011). “Algorithms for nonnegative matrix factorization with the β-divergence”. *Neural computation*, 23(9), pp. 2421–2456 (cited on pp. 78, 79).

Finch, A. and Sumita, E. (2010). “Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model”. In: *Proceedings of the 2010 Named Entities Workshop*, pp. 48–52 (cited on p. 45).

Finot, L. (1995). *Études épigraphiques sur le pays Cham*. 7. Ecole Francaise D’Extreme Orient (cited on p. 3).

Fischer, A., Frinken, V., Fornés, A., and Bunke, H. (2011). “Transcription alignment of Latin manuscripts using hidden Markov models”. In: *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, pp. 29–36 (cited on p. 11).

Fujii, A. and Ishikawa, T. (2001). “Japanese/English cross-language information retrieval: Exploration of query translation and transliteration”. *Computers and the Humanities*, 35(4), pp. 389–420 (cited on p. 45).

Gatos, B. (2014). *Imaging Techniques in Document Analysis Processes*. (Cited on p. 63).

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). “Convolutional sequence to sequence learning”. In: *International conference on machine learning*. PMLR, pp. 1243–1252 (cited on p. 128).

Glauberman, M. H. (1956). “Character recognition for business machines”. *Electronics*, 29(2), pp. 132–136 (cited on p. 36).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). “Generative adversarial networks”. *Communications of the ACM*, 63(11), pp. 139–144 (cited on pp. 16, 21).

Grant, A., Sidwell, P., et al. (2005). *Chamic and beyond: studies in mainland Austronesian languages*. Pacific Linguistics, Research School of Pacific and Asian Studies, The ... (cited on p. 52).

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376 (cited on p. 40).

Griffiths, A., Lepoutre, A., Southworth, W. A., and Phàn, T. (2012). “The Inscriptions of Campā at the Museum of Cham Sculpture in Đà Nang”. *Hanoi: EFEO, Ho Chi Minh City: Center for Vietnamese and Southeast Asian Studies and VNUHCM Publishing House* (cited on p. 3).

Grüning, T., Labahn, R., Diem, M., Kleber, F., and Fiel, S. (2018). “Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents”. In: *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, pp. 351–356 (cited on p. 104).

Grüning, T., Leifert, G., Strauß, T., Michael, J., and Labahn, R. (2019). "A two-stage method for text line detection in historical documents". *International Journal on Document Analysis and Recognition (IJDAR)*, 22(3), pp. 285–302 (cited on pp. 33, 34, 112).

Gruuening, T., Leifert, G., Strauss, T., and Labahn, R. (2017). "A robust and binarization-free approach for text line detection in historical documents". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. IEEE, pp. 236–241 (cited on p. 47).

He, J., Do, Q., Downton, A. C., and Kim, J. (2005). "A comparison of binarization methods for historical archive documents". In: *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, pp. 538–542 (cited on p. 12).

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Deep residual learningfor image recognition". *CoRR, abs/1512*, 3385, p. 2 (cited on pp. 39, 40, 67, 72, 126).

– (2016). "Identity mappings in deep residual networks". In: *European conference on computer vision*. Springer, pp. 630–645 (cited on p. 137).

He, S. and Schomaker, L. (2019). "DeepOtsu: Document enhancement and binarization using iterative deep learning". *Pattern recognition*, 91, pp. 379–390 (cited on p. 16).

Held, M. and Karp, R. M. (1971). "The traveling-salesman problem and minimum spanning trees: Part II". *Mathematical programming*, 1(1), pp. 6–25 (cited on p. 102).

Hennis, R. (1968). "The IBM 1975 optical page reader Part I: System design". *IBM Journal of Research and Development*, 12(5), pp. 346–353 (cited on p. 24).

Hoffman, R. L. and McCullough, J. W. (1971). "Segmentation methods for recognition of machine-printed characters". *IBM Journal of Research and Development*, 15(2), pp. 153–165 (cited on p. 24).

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141 (cited on p. 134).

Hu, M.-K. (1962). “Visual pattern recognition by moment invariants”. *IRE transactions on information theory*, 8(2), pp. 179–187 (cited on p. 36).

Husnain, M., Saad Missen, M. M., Mumtaz, S., Jhanidr, M. Z., Coustaty, M., Muzzamil Luqman, M., Ogier, J.-M., and Sang Choi, G. (2019). “Recognition of Urdu handwritten characters using convolutional neural network”. *Applied Sciences*, 9(13), p. 2758 (cited on p. 39).

Hyvärinen, A., Hoyer, P., and Oja, E. (1998). “Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation”. *Advances in Neural Information Processing Systems*, 11 (cited on pp. 18, 19, 65, 78, 79).

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134 (cited on p. 78).

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). “Perceptual losses for real-time style transfer and super-resolution”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, pp. 694–711 (cited on pp. 66, 72).

Kahle, P., Colutto, S., Hackl, G., and Mühlberger, G. (2017). “Transkribus-a service platform for transcription, recognition and retrieval of historical documents”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 4. IEEE, pp. 19–24 (cited on p. 46).

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., and Kavukcuoglu, K. (2016). “Neural machine translation in linear time”. *arXiv preprint arXiv:1610.10099* (cited on p. 128).

Kanezaki, A. (2018). “Unsupervised image segmentation by backpropagation”. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 1543–1547 (cited on p. 86).

Kang, L., Riba, P., Rusiñol, M., Fornés, A., and Villegas, M. (2022). “Pay attention to what you read: non-recurrent handwritten text-line recognition”. *Pattern Recognition*, 129, p. 108766 (cited on pp. 43, 124, 138).

Kang, L., Riba, P., Villegas, M., Forn'es, A., and Rusiñol, M. (2019). "Candidate Fusion: Integrating Language Modelling into a Sequence-to-Sequence Handwritten Word Recognition Architecture". *ArXiv*, abs/1912.10308 (cited on p. 41).

Kang, L., Toledo, J. I., Riba, P., Villegas, M., Fornés, A., and Rusiñol, M. (2018). "Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition". In: *German Conference on Pattern Recognition* (cited on pp. 41, 125, 127, 138, 140).

Kass, D. and Vats, E. (2022). "AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks". *ArXiv*, abs/2201.09390 (cited on p. 41).

Kasturi, R., O'gorman, L., and Govindaraju, V. (2002). "Document image analysis: A primer". *Sadhana*, 27, pp. 3–22 (cited on p. 9).

Kesiman, M. W. A. (2018). "Document image analysis of Balinese palm leaf manuscripts: Analyse d'images de documents des manuscrits balinais sur feuilles de palmier". PhD thesis. La Rochelle (cited on pp. 10, 11, 13, 45, 60, 89, 104).

Kesiman, M. W. A., Burie, J.-C., and Ogier, J.-M. (2016a). "A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript". In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 325–330 (cited on pp. 25, 31).

Kesiman, M. W. A., Prum, S., Burie, J.-C., and Ogier, J.-M. (2016b). "Study on feature extraction methods for character recognition of Balinese script on palm leaf manuscript images". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 4017–4022 (cited on p. 38).

Khurshid, K., Siddiqi, I., Faure, C., and Vincent, N. (2009). "Comparison of Niblack inspired binarization methods for ancient documents". In: *Document Recognition and Retrieval XVI*. Vol. 7247. SPIE, pp. 267–275 (cited on pp. 12, 14, 39).

Kiessling, B. (2020). "A modular region and text line layout analysis system". In: *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 313–318 (cited on p. 112).

Kiessling, B., Tissot, R., Stokes, P., and Ezra, D. S. B. (2019). "eScriptorium: an open source platform for historical document analysis". In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 2. IEEE, pp. 19–19 (cited on p. 46).

Kim, K. M., Park, J. J., Song, Y. G., Kim, I. C., and Suen, C. Y. (2004). "Recognition of handwritten numerals using a combined classifier with hybrid features". In: *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004. Proceedings*. Springer, pp. 992–1000 (cited on pp. 36, 38).

Kim, W., Kanezaki, A., and Tanaka, M. (2020). "Unsupervised learning of image segmentation based on differentiable feature clustering". *IEEE Transactions on Image Processing*, 29, pp. 8055–8068 (cited on p. 86).

Kirsch, R. A. (1971). "Computer determination of the constituent structure of biological images". *Computers and biomedical research*, 4(3), pp. 315–328 (cited on pp. 36, 37).

Kise, K. (2014). "Page segmentation techniques in document analysis". In: *Handbook of Document Image Processing and Recognition*. Springer, pp. 135–175 (cited on p. 24).

Klink, S. and Kieninger, T. (2001). "Rule-based document structure understanding with a fuzzy combination of layout and textual features". *International Journal of Document Analysis and Recognition*, 4(1), pp. 18–26 (cited on p. 22).

Krishnamoorthy, M., Nagy, G., Seth, S., and Viswanathan, M. (1993). "Syntactic segmentation and labeling of digitized pages from technical journals". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7), pp. 737–747 (cited on p. 22).

Krull, A., Buchholz, T., and Jug, F. (2018). "Noise2Void - Learning Denoising from Single Noisy Images". *CoRR*, abs/1811.10980 (cited on p. 21).

Kumar, S. (2010). "Neighborhood pixels weights-a new feature extractor". *International Journal of Computer Theory and Engineering*, 2(1), p. 69 (cited on pp. 36, 37).

Lebrun, M. and Leclaire, A. (2012). "An implementation and detailed analysis of the K-SVD image denoising algorithm". *Image Processing On Line*, 2, pp. 96–133 (cited on p. 18).

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, 86(11), pp. 2278–2324 (cited on p. 39).

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. (2018). "Noise2Noise: Learning Image Restoration without Clean Data". *CoRR*, abs/1803.04189 (cited on p. 21).

Lemaitre, A., Camillerapp, J., and Coüasnon, B. (2008). "Multiresolution cooperation makes easier document structure recognition". *International Journal of Document Analysis and Recognition (IJDAR)*, 11(2), pp. 97–109 (cited on p. 22).

Levi, G. and Montanari, U. (1970). "A grey-weighted skeleton". *Information and Control*, 17(1), pp. 62–91 (cited on pp. 94, 96).

Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., and Wei, F. (2021). "Trocr: Transformer-based optical character recognition with pre-trained models". *arXiv preprint arXiv:2109.10282* (cited on pp. 42, 141, 142).

Liao, P.-S., Chen, T.-S., Chung, P.-C., et al. (2001). "A fast algorithm for multilevel thresholding". *J. Inf. Sci. Eng.*, 17(5), pp. 713–727 (cited on p. 13).

Likforman-Sulem, L., Darbon, J., and Smith, E. H. B. (2011). "Enhancement of historical printed document images by combining total variation regularization and non-local means filtering". *Image and vision computing*, 29(5), pp. 351–363 (cited on p. 10).

Likforman-Sulem, L., Zahour, A., and Taconet, B. (2007). "Text line segmentation of historical documents: a survey". *International Journal of Document Analysis and Recognition (IJDAR)*, 9, pp. 123–138 (cited on pp. 26, 87, 88).

Lin, X. and Xiong, Y. (2006). "Detection and analysis of table of contents based on content association". *International Journal on Document Analysis and Recognition*, 8(2-3), p. 132 (cited on p. 22).

Liu, C.-L., Nakashima, K., Sako, H., and Fujisawa, H. (2003). "Handwritten digit recognition: benchmarking of state-of-the-art techniques". *Pattern recognition*, 36(10), pp. 2271–2285 (cited on p. 39).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). "Roberta: A robustly optimized bert pretraining approach". *arXiv preprint arXiv:1907.11692* (cited on p. 142).

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440 (cited on pp. 33, 34).

Lu, Y. (1993). "On the segmentation of touching characters". In: *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*. IEEE, pp. 440–443 (cited on p. 25).

Luong, M.-T., Pham, H., and Manning, C. D. (2015a). "Effective approaches to attention-based neural machine translation". *arXiv preprint arXiv:1508.04025* (cited on p. 66).

Luong, M., Pham, H., and Manning, C. D. (2015b). "Effective Approaches to Attention-based Neural Machine Translation". *CoRR*, abs/1508.04025 (cited on p. 126).

Ly, N. T., Nguyen, C. T., and Nakagawa, M. (2019). "An attention-based end-to-end model for multiple text lines recognition in japanese historical documents". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, pp. 629–634 (cited on pp. 43, 44).

Lyu, P., Zhang, C., Liu, S., Qiao, M., Xu, Y., Wu, L., Yao, K., Han, J., Ding, E., and Wang, J. (2022). "MaskOCR: Text Recognition with Masked Encoder-Decoder Pretraining". *arXiv preprint arXiv:2206.00311* (cited on p. 42).

Manmatha, R. and Srimal, N. (1999). "Scale Space Technique for Word Segmentation in Handwritten Documents". In: *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*. SCALE-SPACE '99. Berlin, Heidelberg: Springer-Verlag, 22–33 (cited on p. 26).

Manmatha, R. and Srimal, N. (2002). "Scale space technique for word segmentation in handwritten documents". In: *Scale-Space Theories in Computer Vision: Second International Conference, Scale-Space'99 Corfu, Greece, September 26–27, 1999 Proceedings*. Springer, pp. 22–33 (cited on p. 26).

Mao, X., Shen, C., and Yang, Y.-B. (2016). "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections". *Advances in neural information processing systems*, 29 (cited on pp. 21, 66, 67).

Mechi, O., Mehri, M., Ingold, R., and Essoukri Ben Amara, N. (2019). "Text Line Segmentation in Historical Document Images Using an Adaptive U-Net Architecture". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 369–374 (cited on pp. 33, 34).

Meng, F., Lu, Z., Wang, M., Li, H., Jiang, W., and Liu, Q. (2015). "Encoding source language with convolutional neural network for machine translation". *arXiv preprint arXiv:1503.01838* (cited on p. 128).

Michael, J., Labahn, R., Grüning, T., and Zöllner, J. (2019). "Evaluating sequence-to-sequence models for handwritten text recognition". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, pp. 1286–1293 (cited on p. 41).

Monnier, T. and Aubry, M. (2020). "docExtractor: An off-the-shelf historical document element extraction". In: *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 91–96 (cited on p. 104).

Mori, S., Suen, C. Y., and Yamamoto, K. (1992). "Historical review of OCR research and development". *Proceedings of the IEEE*, 80(7), pp. 1029–1058 (cited on p. 36).

Na, B., Kim, Y., and Park, S. (2022). "Multi-modal text recognition networks: Interactive enhancements between visual and semantic features". In: *European Conference on Computer Vision*. Springer, pp. 446–463 (cited on pp. 42, 43, 141, 142).

Nagy, G., Kanai, J., Krishnamoorthy, M., Thomas, M., and Viswanathan, M. (2000). "Two complementary techniques for digitized document analysis". In: *Proceedings of the ACM conference on Document processing systems*, pp. 169–176 (cited on p. 23).

Nguyen, C. K., Nguyen, C. T., and Masaki, N. (2017). "Tens of thousands of nom character recognition by deep convolution neural networks". In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, pp. 37–41 (cited on p. 39).

Nguyen, M.-T., Schweyer, A.-V., Le, T.-L., Tran, T.-H., and Vu, H. (2019a). "Improving Ancient Cham Glyph Recognition from Cham Inscription Images Using Data Augmentation and Transfer Learning". In: *New Trends in Image Analysis and Processing – ICIAP 2019*. Ed. by M. Cristani, A. Prati, O. Lanz, S. Messelodi, and N. Sebe. Cham: Springer International Publishing, pp. 115–125 (cited on p. 38).

Nguyen, M.-T., Shweyer, A.-V., Le, T.-L., Tran, T.-H., and Vu, H. (2019b). "Preliminary Results on Ancient Cham Glyph Recognition from Cham Inscription images". In: *2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6 (cited on pp. 24, 38, 55).

Nguyen, N., Nguyen, T., Tran, V., Tran, M.-T., Ngo, T. D., Nguyen, T. H., and Hoai, M. (2021a). "Dictionary-guided scene text recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7383–7392 (cited on p. 41).

Nguyen, T.-N., Burie, J.-C., Le, T.-L., and Schweyer, A.-V. (2021b). "On the Use of Attention in Deep Learning Based Denoising Method for Ancient Cham Inscription Images". In: *Document Analysis and Recognition – ICDAR 2021*. Ed. by J. Lladós, D. Lopresti, and S. Uchida. Cham: Springer International Publishing, pp. 400–415 (cited on p. 7).

Nguyen, T.-N., Burie, J.-C., Le, T.-L., and Schweyer, A.-V. (2022). "An effective method for text line segmentation in historical document images". In: *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 1593–1599 (cited on p. 7).

Niblack, W. (1985). *An introduction to digital image processing*. Strandberg Publishing Company (cited on pp. 14, 78).

Nicolaou, A. and Gatos, B. (2009). "Handwritten Text Line Segmentation by Shredding Text into its Lines". In: *2009 10th International Conference on Document Analysis and Recognition*, pp. 626–630 (cited on p. 30).

Nomura, S., Yamanaka, K., Shiose, T., Kawakami, H., and Katai, O. (2009). "Morphological preprocessing method to thresholding degraded word images". *Pattern Recognition Letters*, 30(8), pp. 729–744 (cited on p. 10).

O'Gorman, L. (1993). "The document spectrum for page layout analysis". *IEEE Transactions on pattern analysis and machine intelligence*, 15(11), pp. 1162–1173 (cited on pp. 22, 25).

O'Gorman, L. and Kasturi, R. (1995). *Document image analysis*. Vol. 39. IEEE Computer Society Press Los Alamitos (cited on p. 22).

Oliveira, S. A., Seguin, B., and Kaplan, F. (2018). "dhSegment: A generic deep-learning approach for document segmentation". In: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 7–12 (cited on p. 104).

Otsu, N. (1979). "A threshold selection method from gray-level histograms". *IEEE transactions on systems, man, and cybernetics*, 9(1), pp. 62–66 (cited on pp. 13, 78).

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318 (cited on p. 135).

Park, J., Woo, S., Lee, J.-Y., and Kweon, I. S. (2018). "Bam: Bottleneck attention module". *arXiv preprint arXiv:1807.06514* (cited on pp. 69, 70).

Pastor-Pellicer, J., España-Boquera, S, Zamora-Martínez, F., Afzal, M. Z., and Castro-Bleda, M. J. (2015). "Insights on the use of convolutional neural networks for document image binarization". In: *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part II 13*. Springer, pp. 115–126 (cited on pp. 14, 15).

Peng, X., Cao, H., and Natarajan, P. (2017). "Using convolutional encoder-decoder for document image binarization". In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE, pp. 708–713 (cited on p. 15).

Phuong, T. K. and Lockhart, B. (2011). *The Cham of Vietnam: History, society and art*. NUS Press (cited on p. 55).

Pitas, I. and Venetsanopoulos, A. N. (2013). *Nonlinear digital filters: principles and applications*. Vol. 84. Springer Science & Business Media (cited on p. 16).

Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P. (2003). “Image denoising using scale mixtures of Gaussians in the wavelet domain”. *IEEE Transactions on Image processing*, 12(11), pp. 1338–1351 (cited on pp. 16, 18).

Pratikakis, I., Zagoris, K., Barlas, G., and Gatos, B. (2016). “ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016)”. In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 619–623 (cited on p. 104).

Qiao, Z., Zhou, Y., Wei, J., Wang, W., Zhang, Y., Jiang, N., Wang, H., and Wang, W. (2021). “PIMNet: a parallel, iterative and mimicking network for scene text recognition”. In: *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2046–2055 (cited on p. 43).

Ram, I., Elad, M., and Cohen, I. (2011). “Generalized tree-based wavelet transform”. *IEEE Transactions on Signal Processing*, 59(9), pp. 4199–4209 (cited on pp. 16, 18).

Rasband, W. (2011). “US National Institutes of Health”. *http://imagej. nih. gov/ij/* (cited on p. 58).

Renton, G., Chatelain, C., Adam, S., Kermorvant, C., and Paquet, T. (2017). “Handwritten text line segmentation using fully convolutional network”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 5. IEEE, pp. 5–9 (cited on pp. 33, 34, 88).

Ronneberger, O., Fischer, P., and Brox, T. (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, pp. 234–241 (cited on pp. 34, 66, 67, 78).

Rudin, L. I., Osher, S., and Fatemi, E. (1992). “Nonlinear total variation based noise removal algorithms”. *Physica D: nonlinear phenomena*, 60(1-4), pp. 259–268 (cited on pp. 18, 65, 78).

Sauvola, J. and Pietikäinen, M. (2000). “Adaptive document image binarization”. *Pattern recognition*, 33(2), pp. 225–236 (cited on pp. 14, 78).

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., and Rueckert, D. (2019). "Attention gated networks: Learning to leverage salient regions in medical images". *Medical image analysis*, 53, pp. 197–207 (cited on p. 68).

Schweyer, A.-V. (1999). "Chronologie des inscriptions publiées du Campā: Études d'épigraphie cam–I". *Bulletin de l'École française d'Extrême-Orient*, pp. 321–344 (cited on p. 3).

Seuret, M., Ben Ezra, D. S., and Liwicki, M. (2017). "Robust Heartbeat-Based Line Segmentation Methods for Regular Texts and Paratextual Elements". In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*. HIP2017. Kyoto, Japan: Association for Computing Machinery, 71–76 (cited on pp. 27, 118).

Shamsher, I., Ahmad, Z., Orakzai, J. K., and Adnan, A. (2007). "OCR for printed Urdu script using feed forward neural network". In: *Proceedings of World Academy of Science, Engineering and Technology*. Vol. 23, pp. 172–175 (cited on p. 39).

Shi, B., Bai, X., and Yao, C. (2015). "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition". *CoRR*, abs/1507.05717 (cited on pp. 39, 40, 124, 138, 141, 142).

Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., and Bai, X. (2018). "Aster: An attentional scene text recognizer with flexible rectification". *IEEE transactions on pattern analysis and machine intelligence*, 41(9), pp. 2035–2048 (cited on p. 40).

Shi, M., Fujisawa, Y., Wakabayashi, T., and Kimura, F. (2002). "Handwritten numeral recognition using gradient and curvature of gray scale image". *Pattern Recognition*, 35(10), pp. 2051–2059 (cited on p. 36).

Shi, Z., Setlur, S., and Govindaraju, V. (2005). "Text extraction from gray scale historical document images using adaptive local connectivity map". In: *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, pp. 794–798 (cited on p. 31).

– (2009). "A steerable directional local profile technique for extraction of handwritten arabic text lines". In: *2009 10th International Conference on Document Analysis and Recognition*. IEEE, pp. 176–180 (cited on pp. 31, 32).

Shishtla, P. M., Ganesh, S., Subramaniam, S., and Varma, V. (2009). "A language-independent transliteration schema using character aligned models at NEWS 2009". In: *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pp. 40–43 (cited on p. 45).

Simistira, F., Bouillon, M., Seuret, M., Würsch, M., Alberti, M., Ingold, R., and Liwicki, M. (2017). "ICDAR2017 Competition on Layout Analysis for Challenging Medieval Manuscripts". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 01, pp. 1361–1370 (cited on pp. 7, 11, 93, 104, 107, 109, 119).

Simonyan, K. and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition". *arXiv preprint arXiv:1409.1556* (cited on pp. 39, 72, 126, 137, 138).

Singh, S. S. and Karayev, S. (2021b). "Full page handwriting recognition via image to sequence extraction". In: *International Conference on Document Analysis and Recognition*. Springer, pp. 55–69 (cited on pp. 43, 44).

Singh, S. S. and Karayev, S. (2021a). "Full Page Handwriting Recognition via Image to Sequence Extraction". *CoRR*, abs/2103.06450 (cited on p. 25).

Smith, R. (2007). "An overview of the Tesseract OCR engine". In: *Ninth international conference on document analysis and recognition (ICDAR 2007)*. Vol. 2. IEEE, pp. 629–633 (cited on p. 35).

Souibgui, M. A. and Kessentini, Y. (2020). "De-gan: A conditional generative adversarial network for document enhancement". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), pp. 1180–1191 (cited on p. 21).

Staelin, C., Elad, M., Greig, D., Shmueli, O., and Vans, M. (2007). "Biblio: automatic metadata extraction". *International Journal of Document Analysis and Recognition (IJDAR)*, 10, pp. 113–126 (cited on p. 23).

Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). "ICDAR 2013 Handwriting Segmentation Contest". In: *2013 12th International Conference on Document Analysis and Recognition*, pp. 1402–1406 (cited on p. 108).

Story, G. A., O'Gorman, L., Fox, D., Schaper, L. L., and Jagadish, H. (1992). "The RightPages image-based electronic library for alerting and browsing". *Computer*, 25(9), pp. 17–26 (cited on p. 22).

Strathy, N. W., Suen, C. Y., and Krzyzak, A. (1993). "Segmentation of handwritten digits using contour features". In: *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*. IEEE, pp. 577–580 (cited on p. 24).

Su, T., Jia, S., Wang, Q., Sun, L., and Wang, R. (2016). "Novel character segmentation method for overlapped Chinese handwriting recognition based on LSTM neural networks". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 1141–1146 (cited on p. 25).

Surinta, O., Holtkamp, M., Karabaa, F., Van Oosten, J.-P., Schomaker, L., and Wiering, M. (2014). "A Path Planning for Line Segmentation of Handwritten Documents". In: *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 175–180 (cited on pp. 29–31, 98, 111, 112).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (cited on p. 72).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826 (cited on p. 132).

Tan, M. and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR, pp. 6105–6114 (cited on pp. 39, 126, 134, 137).

Tang, X., Lai, Y., Liu, Y., Fu, Y., and Fang, R. (2021). "Visual-semantic transformer for scene text recognition". *arXiv preprint arXiv:2112.00948* (cited on p. 131).

Tensmeyer, C. and Martinez, T. (2017). "Document image binarization with fully convolutional neural networks". In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE, pp. 99–104 (cited on p. 16).

Thurgood, G. (1999). *From ancient Cham to modern dialects: two thousand years of language contact and change*. Vol. 28. University of Hawaii press (cited on p. 3).

Tikhonov, A. N. (1963). "On the solution of ill-posed problems and the method of regularization". In: *Doklady akademii nauk*. Vol. 151. 3. Russian Academy of Sciences, pp. 501–504 (cited on p. 18).

Tomasi, C. and Manduchi, R. (1998). "Bilateral filtering for gray and color images". In: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, pp. 839–846 (cited on p. 16).

Toriwaki, J.-i. and Fukumura, T. (1978). "Extraction of structural information from grey pictures". *Computer Graphics and Image Processing*, 7(1), pp. 30–51 (cited on p. 37).

Trier, Ø. D., Jain, A. K., and Taxt, T. (1996). "Feature extraction methods for character recognition-a survey". *Pattern recognition*, 29(4), pp. 641–662 (cited on p. 36).

Valy, D. (2020). "Document image analysis and text recognition on Khmer historical manuscripts." PhD thesis. Catholic University of Louvain, Louvain-la-Neuve, Belgium (cited on pp. 11, 89).

Valy, D., Verleysen, M., and Sok, K. (2016). "Line Segmentation Approach for Ancient Palm Leaf Manuscripts Using Competitive Learning Algorithm". In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 108–113 (cited on p. 26).

– (2017). "Line segmentation for grayscale text images of khmer palm leaf manuscripts". In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, pp. 1–6 (cited on p. 26).

Van Han, P., Edmondson, J., and Gregerson, K. (1997). "Eastern Cham as a tone language". *Mon-Khmer Studies*, 20, pp. 31–43 (cited on p. 52).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention is all you need". *Advances in neural information processing systems*, 30 (cited on pp. 42, 66, 125, 128, 129).

Vo, Q. N., Kim, S. H., Yang, H. J., and Lee, G. S. (2018a). "Text line segmentation using a fully convolutional network in handwritten document images". *IET Image Processing*, 12(3), pp. 438–446 (cited on pp. 33, 34).

Vo, Q. N., Kim, S. H., Yang, H. J., and Lee, G. (2018b). "Binarization of degraded document images based on hierarchical deep supervised network". *Pattern Recognition*, 74, pp. 568–586 (cited on p. 15).

Vogler, N., Allen, J. P., Miller, M. T., and Berg-Kirkpatrick, T. (2021). "Lacuna Reconstruction: Self-supervised Pre-training for Low-Resource Historical Document Transcription". *arXiv preprint arXiv:2112.08692* (cited on p. 42).

Vu, M. T., Le, V. L., and Beurton-Aimar, M. (2021). "IHR-NomDB: The Old Degraded Vietnamese Handwritten Script Archive Database". In: *International Conference on Document Analysis and Recognition*. Springer, pp. 85–99 (cited on p. 11).

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). "High-resolution image synthesis and semantic manipulation with conditional gans". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807 (cited on pp. 67, 78).

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). "Image quality assessment: from error visibility to structural similarity". *IEEE transactions on image processing*, 13(4), pp. 600–612 (cited on p. 76).

Wen, Y., Lu, Y., and Shi, P. (2007). "Handwritten Bangla numeral recognition system and its application to postal automation". *Pattern recognition*, 40(1), pp. 99–107 (cited on p. 36).

Wiener, N., Wiener, N., Mathematician, C., Wiener, N., Wiener, N., and Mathématicien, C. (1949). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Vol. 113. 21. MIT press Cambridge, MA (cited on p. 16).

Wigington, C., Tensmeyer, C., Davis, B., Barrett, W., Price, B., and Cohen, S. (2018). "Start, Follow, Read: End-to-End Full-Page Handwriting Recognition". In: *Computer Vision – ECCV 2018*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Cham: Springer International Publishing, pp. 372–388 (cited on pp. 25, 43).

Wolf, C., Jolion, J.-M., and Chassaing, F. (2002). "Text localization, enhancement and binarization in multimedia documents". In: *2002 International Conference on Pattern Recognition*. Vol. 2. IEEE, pp. 1037–1040 (cited on p. 14).

Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). "Document analysis system". *IBM journal of research and development*, 26(6), pp. 647–656 (cited on pp. 30, 31).

Yousef, M. and Bishop, T. E. (2020). "OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by learning to unfold". *CoRR*, abs/2006.07491 (cited on pp. 43, 44).

Yousef, M., Hussain, K. F., and Mohammed, U. S. (2018). "Accurate, Data-Efficient, Unconstrained Text Recognition with Convolutional Neural Networks". *Pattern Recognit.*, 108, p. 107482 (cited on p. 43).

Zahour, A., Taconet, B., Likforman-Sulem, L., and Boussellaa, W. (2009). "Overlapping and multi-touching text-line segmentation by Block Covering analysis". *Pattern analysis and applications*, 12, pp. 335–351 (cited on p. 30).

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising". *IEEE transactions on image processing*, 26(7), pp. 3142–3155 (cited on pp. 21, 66).

Zhang, K., Zuo, W., and Zhang, L. (2018). "FFDNet: Toward a fast and flexible solution for CNN-based image denoising". *IEEE Transactions on Image Processing*, 27(9), pp. 4608–4622 (cited on p. 21).

Zhao, J., Shi, C., Jia, F., Wang, Y., and Xiao, B. (2019). "Document image binarization with cascaded generators of conditional generative adversarial networks". *Pattern Recognition*, 96, p. 106968 (cited on p. 16).

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232 (cited on p. 16).

Zou, Y., Liu, Y., Liu, Y., and Wang, K. (2011). “Overlapped handwriting input on mobile phones”. In: *2011 International Conference on Document Analysis and Recognition*. IEEE, pp. 369–373 (cited on p. 25).