FACULTY OF ENGINEERING
AND ARCHITECTURE

Unlocking the Potential of Digital Archives via Artificial Intelligence

Kenzo Milleville

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors

Prof. Steven Verstockt, PhD* - Prof. Nico Van de Weghe, PhD**

* Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

** Department of Geography
Faculty of Sciences, Ghent University

December 2023

GHENT
UNIVERSITY

ISBN 978-94-6355-792-4
NUR 984
Wettelijk depot: D/2023/10.500/124

Members of the Examination Board

Chair

Prof. Patrick De Baets, PhD, Ghent University

Other members entitled to vote

Prof. Dieter De Witte, PhD, Ghent University
Prof. Haosheng Huang, PhD, Ghent University
Prof. Aleksandra Pizurica, PhD, Ghent University
Prof. Thomas Smits, PhD, Universiteit van Amsterdam, the Netherlands
Prof. Christophe Verbruggen, PhD, Ghent University

Supervisors

Prof. Steven Verstockt, PhD, Ghent University
Prof. Nico Van de Weghe, PhD, Ghent University

Acknowledgements

My interest and passion for artificial intelligence started during my high school period. The progress and advances big tech companies like Google were making with AI, inspired me to pursue a degree in computer science. During my degree, I attended lessons in computer vision from Steven Verstockt, which was the first time I was introduced to Python and OpenCV. I immediately knew this was what I wanted to do and later joined IDLab and Cartogis for a PhD in computer science, with promotors Steven Verstockt and Nico Van de Weghe. I was immediately welcomed by both research groups and started my career as a PhD student.

I want to thank Steven and Nico for their continued guidance and support during these past five years. Their insights and critiques furthered my critical thinking and expertise in both domains. They were always available for a quick chat or an in-depth discussion, for which I am grateful. I also want to thank my colleagues during these past five years. Starting with Florian Vandecasteele, who was my mentor at the start of my career. He was always there to assist with technical or administrative issues (of which there were a lot in the start) and always knew what to do. Together with Krishna and Jelle, we formed the small but mighty *S-team* at IDLab. We were later joined by Alec, Dilawar, Jelle (aka other Jelle), Dieter, Maarten, Joachim, Robbe, Lore, Ravi, and Martin. I would also like to thank my colleagues from the geography department, Samuel, Laure, Robbe, Jana, Lars, and many more, who assisted me with their geo-skills and welcomed me at the S8 in the Sterre (where I got lost on many occasions).

My colleagues always made it worthwhile to come to the office, even though I hated the transit from Gullegem to Ghent. We would chat about politics, crack jokes constantly, and sometimes even have insightful discussions. Even though I was sometimes working alone on different projects, I never felt alone and could always count on them to help me out. I'll never forget all the amazing experiences I've had at conferences or team-building events and the great people I have met over the years.

Furthermore, I would like to thank my mom and dad, for always believing in me and supporting my goals. I can vividly remember my mom being angry when I completely failed my first year in college. Her response was: You may retake your first year, but if you fail again you'll have to start packing your *brooddoos* and go work for a living! This was the motivation I needed and I passed every exam since then. For this guidance and much more, I will be forever in their debt.

Finally, I'd like to thank my friends and family for their camaraderie and unwavering support, with Youri in particular, who's been my best friend for over a decade now. These past years have flown by, with ups and downs, but my friends & family made sure we had more good times than bad. For all of the above and much more, I am grateful and happy looking back. But now we must look forward and remember: **You don't have to be great to start, but you have to start to be great**.

Gullegem, December 2023
Kenzo Milleville

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

AI	Artificial Intelligence
AOI	Area Of Interest
AP	Average Precision
API	Application Programming Interface

B

BERT	Bidirectional Encoder Representations from Transformers

C

CLIP	Contrastive Language–Image Pre-training
CNN	Convolutional Neural Network
CSV	Comma Separated Values

F

FLOPS Floating-point Operations

FPS Frames Per Second

G

GIS Geographic Information System

GPS Global Positioning System

GPT Generative Pre-trained Transformers

GPX GPS Exchange Format

GDPR General Data Protection Regulation

I

IoU Intersection over Union

J

JSON JavaScript Object Notation

K

KDE Kernel Density Estimation

L

LBS Location-Based Service
LLM Large Language Model
LOD Linked Open Data

M

mAP mean Average Precision
ML Machine Learning

N

NLP Natural Language Processing
NER Named Entity Recognition

O

OCR Optical Character Recognition

OSM OpenStreetMap

R

RANSAC Random sample consensus

R-CNN Region based Convolutional Neural Network

ROI Region Of Interest

U

USGS United States Geological Survey

V

ViT Vision Transformer

Y

YOLO You Only Look Once

Samenvatting
– Summary in Dutch –

Gedurende de afgelopen decennia zijn onderzoeks- en erfgoedinstellingen begonnen met het digitaliseren van hun uitgebreide collecties. Deze digitale archieven zijn beschikbaar voor bijna alle onderzoeksvelden en disciplines, zoals natuurwetenschappen, plantkunde, geografie en cultureel erfgoed. Deze digitaliseringsactie heeft deze collecties geopend voor onderzoekers en het brede publiek en zijn over het algemeen toegankelijk via het internet. Digitalisering is een tijdrovend proces, waarbij objecten gescand, gefotografeerd of getranscribeerd worden en vervolgens geannoteerd worden met metadata. Deze metadata is cruciaal voor het organiseren en bevragen van deze archieven, maar het is vaak beperkt en niet gestandaardiseerd over verschillende instellingen.

Deze digitaliseringsinspanningen hebben de shift naar machine learning, deep learning en datagedreven methoden aangewakkerd en versneld. Door neurale netwerken te trainen op gevarieerde datasets, leren de eerste lagen van het netwerk generieke features. Vervolgens kunnen deze modellen snel worden gefinetuned op nieuwe data, door de laatste lagen van het netwerk te hertrainen. In de afgelopen jaren heeft AI-onderzoek zich gericht op het trainen van grotere en complexere modellen, op enorme datasets. Dergelijke modellen worden doorgaans foundation models genoemd. Ze generaliseren efficiënt naar nieuwe data, zelfs in few- of zero-shot omgevingen (via prompting). Bovendien kunnen dergelijke modellen multimodaal worden getraind en toegepast, waarbij zowel visuele als tekstuele informatie wordt gebruikt. Multimodale technieken maken het mogelijk om visuele inhoud te bevragen via natuurlijke taal en vice versa, waardoor de toegankelijkheid van de collectie sterk toeneemt.

Hoewel datagedreven methodes aanzienlijk minder vakkennis vereisen dan traditionele methoden, resteren er veel uitdagingen om ze efficiënt te gebruiken. Eerst moeten de beschikbare objecten worden gedigitaliseerd, wat vaak veel tijd

en kosten vergt. Daarna moeten deze digitale objecten worden geannoteerd met gedetailleerde metadata of labels, om specifieke modellen te finetunen. Vervolgens moeten de datagedreven methodes worden geïmplementeerd en toegepast op de collectie. Dit vereist doorgaans enige programmeerkennis. Ten slotte vereisen de meeste state-of-the-art AI-modellen veel rekenkracht, vooral bij grote collecties. Dergelijke infrastructuur is vaak niet beschikbaar, wat de implementatie van grote AI-modellen belemmert.

Tijdens mijn onderzoek heb ik gewerkt op vele interdisciplinaire projecten en collecties, met zowel tekst als beeldmateriaal. Het onderzoek was gericht op vier verschillende types collecties: fotoarchieven, rasterkaarten, herbaria en social media data. De meeste van deze collecties hadden één ding gemeen: het gebrek aan gelabelde data. Daarom lag de focus van het onderzoek vooral op hoe we AI-gebaseerde methodes efficiënt kunnen gebruiken om digitale archieven te creëren, analyseren en de bevraging ervan te verbeteren.

Beginnend met fotoarchieven, ontwikkelden en pasten we AI-modellen toe om automatisch belangrijke objecten en locaties te herkennen. Voor alledaagse, courante objecten konden bestaande state-of-the-art detectiemodellen worden ingezet met grote nauwkeurigheid. Om minder vaak voorkomende objecten te detecteren of om de afbeeldingen te geolokaliseren, gebruikten we similarity-based technieken. Om belangrijke personen op de foto's te herkennen, ontwikkelden we een gezichtsherkenningspipeline in samenwerking met Meemoo. We gebruikten open-source modellen en pasten ze toe op meer dan 150.000 afbeeldingen. Via deze pipeline identificeerden we automatisch meer dan 62.000 gezichten uit de archieven met een precisie van 0.936. Bovendien ontwikkelden we een interactieve labelingtool dat meer dan 180.000 labels ontving om de persoonsvoorspellingen te valideren. We kunnen besluiten dat gezichtsherkenningsmodellen nauwkeurig en schaalbaar kunnen worden ingezet op fotoarchieven.

Gedigitaliseerde rasterkaarten waren een ander belangrijk aandachtspunt van ons onderzoek. Historische kaarten zijn vaak de enige bron van betrouwbare geografische informatie, waardoor ze zeer waardevol zijn. Hoewel er al grote hoeveelheden zijn gedigitaliseerd, bevatten ze vaak weinig metadata over welke regio's, plaatsnamen of geografische kenmerken op de kaart zijn afgebeeld. Dergelijke informatie is nodig om de kaarten te integreren in een geografisch informatiesysteem (GIS) of om grootschalige studies uit te voeren. Tijdens ons onderzoek ontwikkelden we een nieuw, automatisch geolokalisatie-algoritme. Eerst worden de plaatsnamen op de kaart gedetecteerd en gegeocodeerd. Vervolgens worden de relatieve locaties van de tekstlabels op de kaart en hun geocoding-

resultaten gebruikt om een regio voor de kaart te voorspellen. Deze regio wordt vervolgens iteratief verfijnd via RANSAC. Dit algoritme bleek zeer accuraat te zijn op historische en hedendaagse topografische kaarten.

Nu voor elke kaart een geolocatie is voorspeld, kunnen deze worden geïmporteerd in een GIS. Echter, om enige vorm van analyse op de afgebeelde geografische kenmerken uit te voeren, moeten deze ook worden geëxtraheerd. Daarom hebben we een casestudy uitgevoerd naar de automatische wegextractie van hedendaagse rasterkaarten. Hierbij gebruikten we beschikbare vectordata om de weglabels te genereren. Ons hertraind U-Net segmentatiemodel behaalde een IoU van 0.804 en kon de meeste wegen correct segmenteren. Uiteindelijk willen we deze aanpak generaliseren naar historische kaarten, waar bijna geen gelabelde gegevens beschikbaar zijn.

Een groot deel van ons onderzoek richtte zich ook op de automatische verwerking van herbaria. Deze herbaria, die plantensoorten documenteren die wereldwijd zijn verzameld, vormen de basis van de systematische plantenkunde. Ze zijn verzameld over meerdere eeuwen en worden zorgvuldig bewaard en gearchiveerd. Elk plantspecimen is bevestigd aan een herbariumvel dat typisch informatie bevat over de wetenschappelijke naam van de plant, de verzameldatum, de geografische herkomst en andere relevante details. We hebben een casestudy uitgevoerd om de soort en het genus van de plant te identificeren op basis van fuzzy tekstmatching en hebben laten zien hoe dit kan worden gebruikt om de bestaande metadata te valideren.

In samenwerking met Plantentuin Meise hebben we de archieven van de Universiteit Gent bijgestaan met hun lopend digitaliseringsproces. Dit proces omvat het fotograferen van elk herbariumvel. Naast elk vel werd een gestandaardiseerde kleurenkaart geplaatst om kleur- en groottereferenties te bieden. We hebben de herbariumvellen en kleurenkaarten automatisch gedecteerd en daarna aan elkaar geplaatst. We hebben twee methodes ontwikkeld om de kleurenkaarten te detecteren. De ene was gebaseerd op ORB-features, terwijl de andere gebruik maakte van een roodkleurig papier dat onder de kleurenkaart werd geplaatst. Deze tweede methode was veel nauwkeuriger (97,6%) en versnelde het digitaliseringsproces.

Om een volledig herbariumvel automatisch te analyseren, hebben we een nieuwe instance segmentation dataset van 250 bladen gecreëerd. Via een semi-automatische labelingmethode konden we binaire plantenpixelmaskers extraheren en later valideren. Daarna hebben we de vellen handmatig geannoteerd met vaak voorkomende objecten zoals meetlatten, kleurenkaarten en barcodes. We

hebben deze dataset gebruikt om drie segmentatiemethodes te evalueren. Er werden verschillende binaire plantsegmentatiemodellen getest, waarbij UNet++ de hoogste IoU van 0.951 behaalde. Voor instance segmentation scoorde Mask2Former gemiddeld het best, met een mAP van 78.9. Tot slot werd de segmentatie ook geherformuleerd als een panoptic segmentation probleem, met de plantklasse als een semantische klasse. Een combinatie van YOLOv8 en UNet++ presteerde beter dan het Mask2Former model en kon nauwkeurig de vellen segmenteren. Hoewel deze resultaten veelbelovend waren, is er verder onderzoek en meer gelabelde data nodig om de automatische verwerking van herbaria op te schalen naar een globaal niveau.

Tot slot hebben we aangetoond hoe je grote social media datasets kan verzamelen, verwerken en analyseren. We hebben meer dan 15 miljoen tweets gerelateerd aan natuurrampen en milieuthema's verzameld en gegeocodeerd. Eerst probeerden we eenvoudige sentiment analysemodellen te gebruiken, maar deze leverden geen nieuwe inzichten op. Daarom hebben we een kleine stance detection dataset van 500 tweets gecreëerd en hebben we daar een BERTweet-model op getraind. Het model behaalde een gemiddelde F1-score van 0.67. We verwachten dat deze score aanzienlijk kan worden verbeterd door extra data te labelen. Bovendien hebben we via een casestudy op een Flickr dataset laten zien hoe je automatisch toeristische hotspots kunt bepalen en deze geospatiaal kunt analyseren.

Dit proefschrift heeft een overzicht gegeven van ons onderzoek naar het gebruik van AI-gebaseerde methoden om digitale archiefcollecties te verrijken. We hebben laten zien hoe deze methoden in de praktijk kunnen worden gebruikt, op verschillende collecties en datasets. We kunnen besluiten dat AI-gebaseerde en datagedreven technieken kunnen helpen bij het creëren, analyseren en onderhouden van digitale archieven. Het implementeren van dergelijke technieken kan enorme kosten en tijd besparen. Het levert ook aanvullende metadata op, wat de bevraagbaarheid van de collecties vergroot.

Summary

Over the past few decades, many scientific and heritage institutions have begun digitizing their vast collections. Such digital archives have become commonplace in nearly all research fields and disciplines, such as natural history, botany, geography, cultural heritage, and more. This digitization effort has opened up these collections to researchers and the public, and are generally accessible via the internet. Digitization is a labor-intensive process, involving scanning, photographing, transcribing, and subsequently annotating the objects with metadata. This metadata is crucial for organizing and querying archives, but is often limited and not standardized across different institutions.

These digitization efforts have fueled and accelerated the shift toward machine learning, deep learning, and data-driven methods. While these methods require significantly less expert knowledge than traditional methods, many challenges remain to use them efficiently. First, the available objects need to be digitized, which often involves a lot of manual effort. Second, these digital objects need to be annotated with detailed metadata or labels, to fine-tune specialized models. Third, the data-driven methods need to be implemented and applied to the collection. This typically requires some programming knowledge. Fourth, most state-of-the-art AI models require a lot of computation, especially on large collections. Such infrastructure is often not available, hindering the implementation of large AI models.

During my research, I have worked on many interdisciplinary projects and collections, featuring both text and images. The research was focused on four different types of collections: photo archives, raster maps, herbaria, and social media data. Most of these collections had one thing in common: the lack of labeled data. Therefore, the research has mainly focused on how we can efficiently use AI-driven methods to process, analyze, and improve the accessibility of digital archives. To achieve this, we utilized robust models that can be quickly fine-tuned to new data. We also improved the data labeling process by shifting the manual

labor from annotation to validation. This drastically reduced the amount of time required, especially for complex annotations like pixel masks.

Starting with archive photo collections, we developed and applied AI models to automatically recognize objects and places of interest. For everyday objects, existing state-of-the-art detection models showed great accuracy and performance. To efficiently detect rare objects or geolocate the images, similarity-based approaches can be used. Furthermore, such models can be trained and applied multimodally, utilizing both vision and language information. These techniques enable querying of the visual content via natural language, further increasing the accessibility of the collection.

To recognize persons of interest in the photographs, we developed a facial recognition pipeline in collaboration with Meemoo. We used open-source models and applied them to over 150,000 images. Via this pipeline, we automatically identified over 62,000 faces from the image archives with a precision of 0.936. Additionally, we developed an interactive labeling tool that received over 180,000 labels to validate the person predictions. We conclude that facial recognition models can be applied accurately and at scale on archive photo collections.

Digitized raster maps were another key focus of our research. Historical maps are often the only source of reliable geographic data, making them very valuable. While large amounts have already been digitized, they often lack information on which regions, place names, or geographical features are depicted on the map. Such information is needed to integrate the maps into a geographical information system (GIS) or to perform large-scale studies. Therefore, we developed a novel, automatic geolocation algorithm. First, the place names on the map were detected and geocoded. Next, the relative location of text labels on the map and their geocoding coordinates were used to estimate a region for the map. This region was then iteratively refined via RANSAC. This algorithm proved very accurate on historical and contemporary topographic maps.

Now that each map has a predicted geolocation, these can be imported into a GIS. However, to perform any kind of analysis on the depicted geographical features, these need to be extracted as well. By using contemporary raster maps and associated vector data, we generated a road segmentation dataset covering the entire Netherlands. Using this dataset, we fine-tuned a U-Net segmentation model, which achieved an IoU of 0.804. The model was fast and could segment most roads accurately. Ultimately, we would like to generalize this approach to historical maps using domain adaptation techniques.

A large portion of our research also focused on the automated processing of

herbarium specimens. These herbarium specimens record plant occurrences collected from all corners of the world, forming the foundation of systematic botany. They have been collected over several centuries and are carefully archived and preserved. Each specimen is attached to a herbarium sheet that typically contains information on the plant's scientific name, collection date, geographical origin, and other relevant details. We performed a case study to identify the plant's species and genus, based on fuzzy text matching and showed how this could be used to validate the existing metadata.

In collaboration with Meise Botanic Garden, we assisted the Ghent University archives with their ongoing digitization process. This process involved photographing each herbarium sheet. A color chart was placed next to each sheet, to provide color and size references. We automatically extracted the herbarium sheets and color cards, which were then stitched together. We developed two methods to detect the color cards. One was based on ORB features, while the other involved placing a red-colored paper under the color card. This second method was far more accurate (97.6%) and greatly sped up the digitization process.

To fully analyze an entire herbarium sheet via computer vision, we needed to create a novel instance segmentation dataset. Via a semi-automatic labeling approach, we extracted the plant masks and later validated them. Then, we manually annotated the sheets with common objects such as rulers, color cards, and barcodes. We used this dataset of 250 sheets to evaluate three segmentation approaches. Different binary plant segmentation models were tested, with UNet++ achieving the highest IoU of 0.951. For instance segmentation, Mask2Former scored best overall with a mAP of 78.9. Finally, the segmentation task was reformulated as a panoptic segmentation problem, with the plant class as a semantic class. A combination of YOLOv8 and UNet++ outperformed the Mask2Former model and could accurately segment the entire sheet. While these results were promising, further research and labeled data are needed to improve and evaluate the automated processing of herbarium specimens on a global scale.

Lastly, we've demonstrated how to collect, process, and analyze large social media datasets. We collected and geolocated over 15 million tweets, related to natural disasters and environmental topics. First, we tried using simple sentiment analysis models, but they did not provide novel insights. Therefore, we created a small stance detection dataset of 500 tweets and retrained a BERTweet model. The model achieved an average F1-score of 0.67. We expect that this score can be improved significantly by labeling additional data. Additionally, via a case study on a Flickr dataset, we've shown how to automatically determine tourism

hotspots and analyze these geospatially. In essence, we've shown that NLP methods can be used to efficiently generate insights on large social media datasets.

We can conclude that AI-based methods can deliver tremendous value in creating, maintaining, and analyzing digital archives at scale. Our research focused on multiple collections and our methods can be easily adapted to similar collections. While a lack of detailed metadata or labeled data inhibits certain solutions, we've presented ways to overcome these issues and implement practical data-driven techniques.

1 Introduction

"The only source of knowledge is experience"

– Albert Einstein

This chapter situates the performed research, summarizes the main contributions, and outlines the structure of this dissertation. The chapter also lists all of the scientific publications written during my research.

1.1 Context

During the past few decades, many scientific and heritage institutions have begun digitizing their vast collections. Such digital archives have become commonplace in nearly all research fields and disciplines, such as natural history, botany, geography, cultural heritage, and more. This digitization effort has opened up these collections to researchers and the public. They are generally accessible via the internet. This mass digitization was only possible due to the tremendous efforts of researchers, archivists, and volunteers. For instance, pictures and documents need to be scanned or photographed, texts and books need to be transcribed. These digital objects are subsequently annotated with additional metadata.

This metadata typically describes the object, its geographic origin, its author, and other features. This metadata is crucial in organizing and querying digital

archives efficiently. It can be easily integrated into existing search engines, enabling detailed filtering of the collection. The available metadata is often limited to the most important object information and is not fully standardized. This is because different institutions tend to have different digitization processes, even for the same type of data. This heterogeneity hinders the accessibility and interoperability of these digital archives. For instance, image collections are rarely annotated with detailed information on what is depicted in the images, which makes it difficult to search for specific persons, objects, or themes.

Even when new data is created in a digital format, it often lacks high-quality metadata or structure. Such data is created constantly, via digital photography, literature, journalism, or other means. This born-digital data is often shared publicly online. Many digital archives collect or create such data, further increasing the size of their collections. Social media platforms, which store billions of images, texts, and other content are a prime example of born-digital archives. Such platforms could be considered as the archives of the future, storing personal memories and moments of cultural significance.

1.1.1 Shift towards Machine Learning

While these mass digitization efforts have been expensive and time-consuming, access to digitized data has fueled and accelerated the shift toward machine learning, deep learning, and data-driven methods. In the past decade, computer vision and natural language processing (NLP) research have made great strides with the advent of neural networks, access to larger datasets, and more performant GPUs. For computer vision, this trend started in 2012, when AlexNet [1] achieved state-of-the-art accuracy on the ImageNet competition [2] by training a deep convolutional neural network (CNN). Since then, practically every state-of-the-art computer vision model has used convolutional layers. One of the most widely used architectures, ResNet [3], introduced residual connections between successive layers (see Figure 1.1). This enabled the training of larger and deeper networks on larger datasets.

By training these models on huge and varied datasets, the lower layers of the network essentially learn generic image features. Then, these models can be fine-tuned on other labels, by replacing the final layers of the network. This process is called transfer learning and is a common practice, as it drastically cuts down the time and labeled data needed to achieve accurate results.

Throughout the years, model architectures have been continuously evolving, until Transformers [5] redefined the state-of-the-art NLP model architecture. Their key innovation was the self-attention mechanism, which allows the model to focus on different parts of the input data with varying degrees of attention, depending on the context.

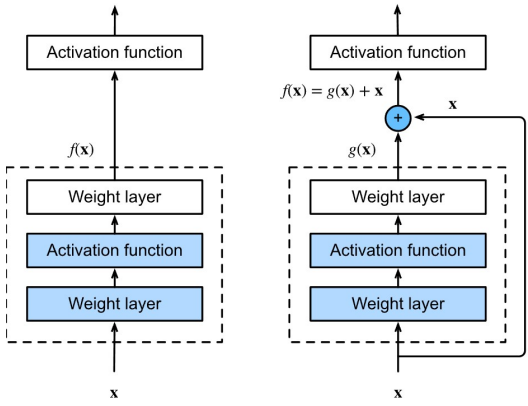


Figure 1.1: A visualization of the residual connections used in the ResNet architecture (Image from [4]).

The input is essentially embedded into a sequence, where each item in the sequence is represented by a query, key, and value. The Transformer then calculates how much attention each item should pay to every other item by multiplying the query of that item with the keys of all other items. This score is then normalized and used to weigh the value (importance) of each item. This enables the model to learn the importance of each item in the context of the whole sequence (e.g. each word in the context of an entire sentence). This generic and efficient architecture has since been adopted by many state-of-the-art models in computer vision and other domains. Figure 1.2 visualizes one of the earlier Vision Transformer (ViT) architectures from [6]. An input image is essentially split into fixed-size patches, embedded with their position embeddings, and fed to a transformer encoder. The attention mechanism then learns the importance of each patch in relation to all other patches.

1.1.2 Current trends in AI

By using (vision) transformers, researchers have been able to train increasingly larger models on increasingly large datasets. These so-called foundation models are typically trained in an unsupervised way on a broad range of data [7]. Once trained, these models can be fine-tuned or adapted to many downstream tasks with little to no supervision. In essence, the models "learn foundational knowledge", on top of which they can quickly learn specialized tasks. This is essentially taking the concept of transfer learning to its extreme, where the models can already achieve a decent performance on new datasets, with minimal (few-shot) or no additional labeled examples (zero-shot).

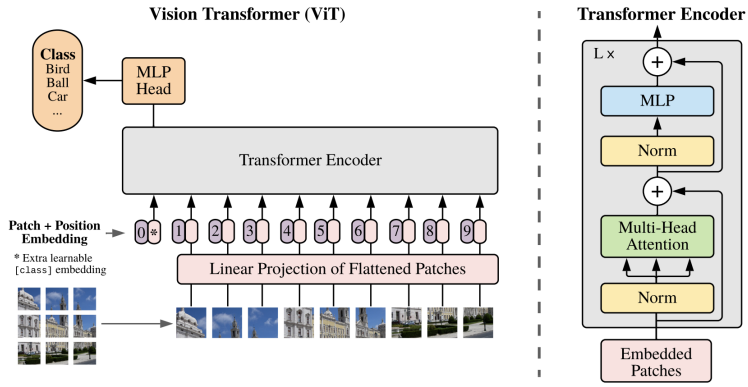


Figure 1.2: Vision transformer architecture from [6]

So, the field of AI has shifted even more towards collecting high-quality data to train these foundation models. Instead of developing specialized models and processing methods for each collection, AI-driven methods can be applied to single or multiple data modalities (multimodal), across collections. For instance, object detection models are applicable to virtually any photograph depicting common objects. Large language models (LLMs) apply to any writing, code, or other textual data, independent of the language or context. Across all media, tremendous progress in automated processing has been made and such automated methods have even surpassed human capabilities in multiple fields [8, 9]. Furthermore, data-driven methods require much less expert knowledge to be used effectively than their traditional counterparts, which require handcrafted features for each specific task.

1.1.3 Lack of Labeled Data

However, for certain research domains, access to high-quality labeled data or pretrained models is limited and is often the main bottleneck in creating scalable solutions. For instance, labeled datasets of digitized herbarium specimens, historical maps, or art collections are scarce and often not fully labeled. This makes it difficult to develop generic and accurate end-to-end models. Many collections containing uncommon objects have this problem. Despite these issues, there has been a sharp increase in new interdisciplinary projects, uniting computer scientists and other domains. These projects typically develop innovative processing methods and publish new labeled datasets, furthering the state-of-the-art and opening up these often-overlooked collections for future research.

Throughout my research, I have worked on many of these interdisciplinary

projects and collections, featuring both text and images. Most of these collections had one thing in common: the lack of labeled data. Therefore, a key focus of our research was how to efficiently use AI methods on collections with limited or no labeled data. To achieve this, we utilized robust models that can be quickly fine-tuned to new data. We also improved the data labeling process by shifting the manual labor from annotation to validation. This drastically reduced the amount of time required, especially for complex annotations like pixel masks. Data-driven methods were the key focus of my research, hence, this dissertation is organized based on the type of collection and data processed.

1.1.4 Research Focus

My Ph.D. was not funded with a scholarship but encompasses my work on multiple research projects over a five-year period. These projects were mainly situated in the cultural heritage and geo domains featuring various partners like Meemoo, GhentCDH, TUWien, KBR, GUM, and Meise Botanic Garden. This allowed me to work on many collections, in multiple domains and gave me a versatile skillset. These projects involved analyzing large collections with little labeled data, forcing me to develop and combine efficient AI and computer vision methods.

This thesis summarizes the research performed on four different types of collections: photo archives, raster maps, herbaria, and textual data, which are visualized in Figure 1.3. Photo collections were enriched with image-level metadata, by recognizing objects and persons of interest. We proposed a novel geolocation algorithm and road segmentation method on raster maps. Herbarium collections were another focus of research. There, we assisted in the digitization and evaluated multiple segmentation methods on a newly created dataset. Lastly, our research on textual data focused on applying NLP methods to large social media datasets and analyzing them geospatially.

The developed methods and research are applicable to any collection containing similar data and can be easily adapted to other collections. To summarize, the general goal of my research and this dissertation can be stated as follows: **How can AI-driven methods be used to automatically process, analyze, and improve the accessibility of digital archives?**

1.2 Outline

Based on the context and problems presented in the previous Section (Section 1.1), Section 1.3, discusses the challenges regarding AI-based processing of digital archives. The first chapter concludes with Sections 1.4 and 1.5, which list the

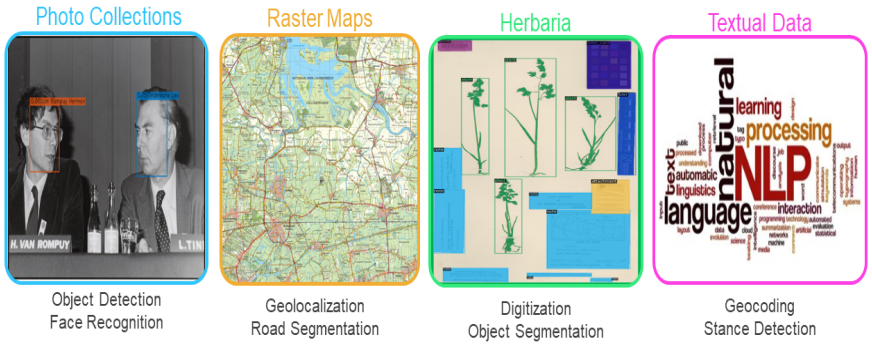


Figure 1.3: Overview of the four different types of collections and data that are discussed in this dissertation.

main research contributions and publications written during my research. Chapter 2 discusses photo collections, and how computer vision can be used effectively to detect objects or persons in these collections. Chapter 3 summarizes the research performed on digitized raster maps. The chapter starts by discussing a novel geolocalization method based on text recognition and geocoding of visible place names. It finishes by presenting road segmentation methods and a case study on walking and cycling maps. In Chapter 4, automated processing methods for digitized herbaria are proposed. These automated methods include preprocessing, dataset creation, specimen identification, and segmentation. Chapter 5 discusses the use of NLP techniques on textual data, with an emphasis on social media data. We show how these techniques can be used to perform geospatial analyses on a global scale. The dissertation concludes in Chapter 6, summarizing the main findings of the research and proposing some future work.

1.3 Challenges

There are many challenges in creating, maintaining, and publishing digital archives. The contents of the collections first need to be digitized and annotated with metadata. Next, the collections need to be easily accessible and searchable. AI-driven methods can assist this process, however, they come with their own challenges.

1.3.1 Digitization

First and foremost, the contents of the collections need to be digitized. This process typically involves the scanning, photographing, or transcription of the phys-

ical objects. This requires a tremendous amount of manual labor, by archive employees, researchers, and volunteers. Many collections across the globe contain thousands to millions of objects. Therefore, it is crucial to minimize the time required to digitize each object. This can be achieved by streamlining the digitization process (developing improved protocols and workflows) and by implementing automated processing and validation methods.

The digitization of the New York Botanical Garden Herbarium is a great example of how new technology can reduce processing time. By incorporating improved equipment and protocols, they doubled the image capture rates per volunteer [10]. This was achieved via a combination of new hardware and integrated software tools, which were much easier to use. Figure 1.4 shows four volunteers using the "Photo-eBox" toolkit to digitize herbarium specimens.

Figure 1.4: Four volunteers using the Photo-eBox toolkit to digitize herbarium specimens (Image from [10]).

While new technologies and workflows have sped up the digitization process, there is always room for improvement. Automatic preprocessing tools will rarely be perfect and sometimes still introduce errors. Therefore, manual correction and validation are still required. However, shifting the manual labor towards validation instead of annotation can greatly speed up the digitization process. For instance, manually removing the background from photographed herbarium specimens is tedious. Via computer vision, we implemented an automated preprocessing method to crop and preprocess the herbarium sheet. The processed

images can then be quickly validated, speeding up the digitization process. Such automated processing methods are also easier to scale than the number of volunteers or employees.

1.3.2 Detailed Metadata

Once the objects are digitized, they are organized within databases or digital repositories to ensure they're accessible to researchers and the general public. During this process, they are annotated with object-level metadata. This metadata provides context about the digital items, offering information on the object's origin, creator, date of creation, subject matter, and much more. Well-curated metadata is crucial in supplying researchers and the public with relevant information. Imagine an archive of photographs without any metadata. Finding photographs depicting a specific person becomes unfeasible, as you would have to search each photo individually. Similarly, having no information about the image contents makes it difficult to find photographs depicting certain locations, themes, or objects.

Annotating objects with detailed metadata entails many challenges. First, it introduces more manual labor, as someone needs to either validate or enter the available information into a digital format. Second, exact metadata might not even be available at all. Looking back to our photo archive, recognizing the visible persons requires intimate knowledge about the collection. Luckily, face recognition models can be used to automatically recognize persons of interest. We successfully used such an approach on a collection of 180,000 archive photographs. Finally, even when metadata is available, there is often no standardized format. Therefore, different institutions and even archivists within a single institution will annotate the same metadata in different ways. This results in inconsistencies within and across collections and hinders their accessibility.

Automatically extracting metadata from historical collections comes with some additional challenges. First, the original data sources can be degraded, generally lowering the accuracy of pre-trained models. Second, many objects depicted in historical photographs are no longer common and will not be detected. Finally, the names of many addresses and places have changed throughout the years. This hinders the automatic matching of recognized text with linked open data sources. To tackle these issues, fuzzy matching and specialized historical gazetteers [11] need to be used. Fuzzy matching enables the linking of non-exact text matches via string similarity metrics and is frequently used to correct text recognition errors.

1.3.3 Labeled Datasets

While AI-based methods are valuable in generating object-level metadata, these models first need to be trained on existing labeled datasets. For common tasks, like object detection on photographs, such labeled datasets already exist (e.g. COCO [12]). These pre-trained models achieve incredible performance without any fine-tuning. This was only made possible by painstakingly annotating thousands to millions of images. For most institutions, this is not feasible and the amount of available metadata or labeled data is limited. This is especially true for collections featuring historical data, like digitized maps, herbarium sheets, books, or other archives.

For such collections, there is generally little to no information on what is depicted in the image. For instance, contemporary maps are generated from available vector datasets using a GIS (Geographic Information System). For historical maps, no such vector data exists. There is no information on the depicted roads, land use, or other geographical features. Therefore, these cannot be easily integrated into a GIS, and large-scale studies cannot be performed without first (manually) processing these maps.

The images in the collection can be annotated with new, task-specific labels. However, this again introduces additional manual labor. For instance, to segment collection-specific objects on images, these first need to be annotated with detailed pixel masks. These masks are difficult and time-intensive to annotate. The same is true for other tasks, like face recognition, which requires a dataset of faces annotated with person names.

Crowdsourcing can help supply large collections with task-specific labels. Via an interactive online platform, volunteers can annotate objects of the collection. Such approaches have been successful in the past, but require additional costs to set up [13, 14]. They also require a set of volunteers willing to help annotate the collections. The resulting labels still need to be validated in some way, as it can be difficult for people unfamiliar with the source material to provide accurate labels. However, for many tasks, computer vision methods can also assist the labeling process.

Instead of manually annotating images with object labels or fine-grained pixel masks, computer vision methods can generate these labels. These will contain many errors and need to be validated. This is a much easier task, as one simply needs to validate each prediction as correct or incorrect. The validated predictions are then used to create a dataset with task-specific labels. This process is called semi-automatic labeling. We used such an approach to create detailed pixel masks of herbarium specimens. Via transfer learning, AI models can then be trained and evaluated on this task. After training, these models can predict new labels for the rest of the collection.

Multimodal and foundation models can help to further reduce the labeling efforts, by providing impressive few or zero-shot accuracy via text or other prompts. For instance, the breakthrough Segment Anything model (SAM) [15], is a generic image segmentation model. It produces object masks, given a point, box, mask, or text input. This enables interactive data labeling, by allowing users to click objects or draw boxes around them. Figure 1.5 visualizes the SAM mask predictions for digitized herbarium sheets, given bounding box prompts. Such novel methods can greatly speed up data labeling and are easy to use. Similar progress has been made for textual data labeling, via large language models. These allow users to specify a text prompt defining the task and the model produces the labels. For instance, you can prompt GPT-4 [16] to label a set of tweets with their sentiment (positive or negative). Studies have already shown that such an approach can outperform crowdsourced workers [17] for a fraction of the cost.

Figure 1.5: SAM mask predictions given bounding box prompts for each specimen.

1.3.4 Lack of Knowledge

AI-based methods can provide tremendous value for digital archives, but someone still needs to implement them. There is often a lack of knowledge on what is already possible with AI to process the collection. Furthermore, many employees or researchers involved with these institutions have little knowledge of how to use

these models. They have limited programming experience, which is generally required to implement AI models. Therefore, interdisciplinary research projects are key in bringing computer scientists together with these institutions, to develop and implement AI-pipelines.

For many problems, there is often an overlap between different research projects. Multiple projects focus on similar problems on their own dataset and generate their own labels and predictions. These datasets are often private and the trained models or code are sometimes not shared. Even when they are publicly available, different projects will use different models, which are generally not interchangeable. For instance, different facial recognition models will generate different embeddings for each face, which can then be matched via a similarity metric. These embeddings cannot be interchanged, so if another project wants to integrate an existing dataset with a new model, the embeddings have to be extracted again.

There has been a positive shift towards more open-source code and models these past few years. Many research projects have started to adopt this as well. Combined with the development of new foundation models, this has greatly reduced the knowledge gap. State-of-the-art models like GPT-4 can be prompted to generate code to train other AI models. While such code might not be optimal, it is already a big step forward to reduce the knowledge gap.

1.3.5 Growing Computational Demands

Besides knowledge and data, AI models also require computing power. Unfortunately, larger and more performant models, typically require more computing power. GPUs are necessary to process large collections in a reasonable amount of time. For most computer vision tasks, low to midrange GPUs will perform quite well, and some optimized models like YOLOv8 [18] or Mobilenets [19] can be run quite fast on CPUs. However, for specific tasks dealing with high-resolution images or complex architectures, more GPU RAM is necessary. This is especially true for state-of-the-art foundation models and LLMs. The larger models also take much longer to train, which is shown in Figure 1.6. It visualizes the estimated amount of petaFLOPS (Floating-point Operations) required to train popular computer vision and language models. While computing power has also increased over the years, the models are growing faster than the speed of available hardware.

In general, there is always a trade-off between processing time and accuracy. The larger and slower models will produce more accurate results but require more compute. Therefore, deciding which model to use on large collections is important. It is often not worth it to use the most cutting-edge models on larger collections. These might be 2% more accurate but require three times the processing time.

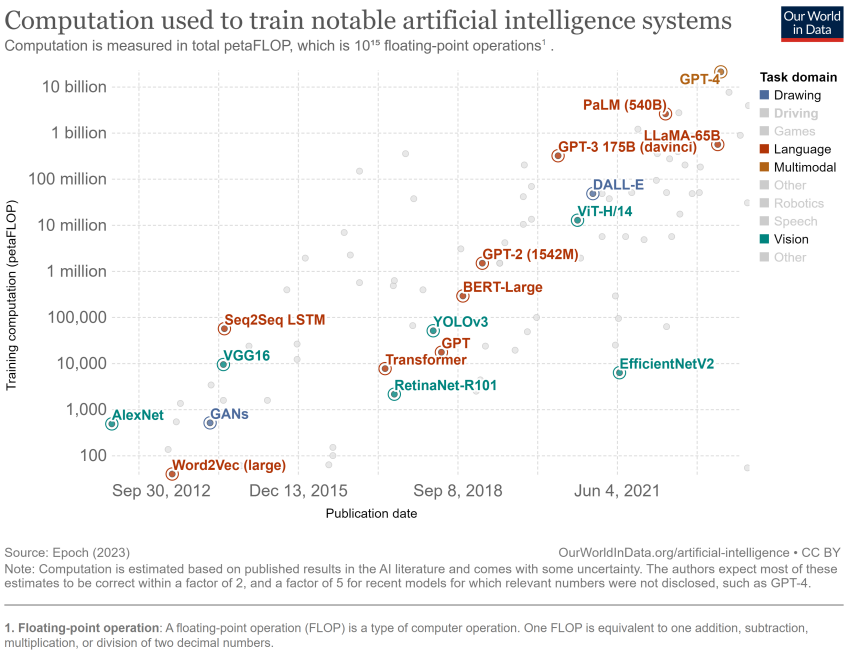


Figure 1.6: Estimated amount of petaFLOPS needed train to popular computer vision and language models (log scale) [20].

This is especially true when the processing needs to be real-time or interactive. Then, smaller and faster models are preferred.

Most institutions do not have the infrastructure necessary to train and run specialized AI models on their collections. There is often little budget for expensive GPU workstations or servers. Therefore, cloud computing providers like AWS and GCP can be used instead. These enable pay-as-you-go pricing and allow you to scale up the required compute as needed. However, this again introduces additional manual effort, as someone needs to set up these cloud servers and data pipelines.

1.3.6 Bias in AI

The last important challenge is that of bias in AI. While this is not studied in detail in this dissertation, it is important to acknowledge the ethical implications and biases inherent to AI systems. These can emerge from various sources, such as the training data, the design of the model, or the context in which it is applied. Face recognition systems are a great example when it comes to such issues. They

are typically less accurate for certain races, perceived genders, or skin tones[1] [2]. Furthermore, they might break privacy laws (e.g. GDPR) or copyright laws without careful use. While it is extremely difficult to overcome these biases, it is vital to be aware of them, especially when using AI models in production.

The processing of historical collections comes with additional challenges. The use of AI might risk reinforcing historical biases or stereotypes encoded in these collections. The outputs of AI models are also typically less accurate on these collections, due to degradation of the source material, changes in language and writing style, but also changes in the visual appearance of objects and people. Clearly, careful analysis and evaluation are required when analyzing sensitive or historical data sources.

1.4 Research contributions

The main contributions of my research can be summarized as follows:

- In collaboration with Meemoo[3], we developed a facial recognition pipeline using open-source models and applied it to over 150,000 images. Using this pipeline, we automatically identified over 62,000 faces from the image archives with a precision of 0.936.
- Developed an online similarity-based labeling tool that received over 180,000 labels to validate the facial recognition pipeline.
- Proposed a novel geolocalization algorithm for topographic raster maps via text recognition and geocoding of the toponyms on the map.
- Evaluated binary and multiclass road segmentation approaches on contemporary topographic maps of the Netherlands, using associated vector data.
- Developed a fully automated preprocessing pipeline, to assist Ghent University in digitizing their collection of 300,000 herbarium specimens.
- For the DISSCO-Flanders[4] project and in collaboration with the botanical garden of Meise[5], we created a novel open-source instance segmentation

[1] https://github.com/deepinsight/insightface/tree/master/python-package

[2] https://storage.googleapis.com/mediapipe-assets/MediaPipe%20BlazeFace%20Model%20Card%20(Short%20Range).pdf

[3] https://meemoo.be/en/news/fame-comes-to-an-end-initial-results

[4] https://dissco-flanders.be/

[5] https://www.plantentuinmeise.be/en

dataset for digitized herbaria. We then evaluated multiple segmentation approaches on this dataset, to accurately segment the plants and common objects on the herbarium sheets with an mAP of 78.9.

- Collected and geolocated over 15 million tweets related to natural disasters and alternative energy sources. Created a novel stance detection dataset and retrained NLP models to analyze this geospatial data on a global scale.

1.5 Publications

1.5.1 Publications in international journals (listed in the Web of Science [6])

1. Groom, Q., **et al.** (2023). *Envisaging a global infrastructure to exploit the potential of digitised collections.* Biodiversity Data Journal, 11, e109439.

2. **Milleville, K.**, Van den broeck A., Vanderperren N., Vissers R., Priem M., Van de Weghe N., & Verstockt S. (2023). *Enriching Image Archives via Facial Recognition.* ACM Journal on Computing and Cultural Heritage.

3. **Milleville, K.**, Chandrasekar, K. K. T., & Verstockt, S. (2023). *Automatic extraction of specimens from multi-specimen herbaria.* ACM Journal on Computing and Cultural Heritage.

4. Ali, D., **Milleville, K.**, Verstockt, S., Van de Weghe, N., Chambers, S., & Birkholz, J. M. (2023). *Computer vision and machine learning approaches for metadata enrichment to improve searchability of historical newspaper collections.* Journal of Documentation.

5. **Milleville, K.**, Verstockt, S., & Van de Weghe, N. (2022). *Automatic Georeferencing of Topographic Raster Maps.* ISPRS International Journal of Geo-Information, 11(7), 387.

[6] The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: Articles included in one of the Web of Science databases 'Science Citation Index', 'Social Science Citation Index' or 'Arts and Humanities Citation Index'. Limited to the publications document type article, review, letter, note, proceedings paper.

1.5.2 Publications in other international journals

1. **Milleville, K.**, Van Ackere, S., Verdoodt, J., Verstockt, S., De Maeyer, P., & Van de Weghe, N. (2023). *Exploring the potential of social media to study environmental topics and natural disasters.* Journal of Location Based Services, 1-15.

1.5.3 Publications in international conferences

1. **Milleville, K.**, Thirukokaranam Chandrasekar, K.K., Van de Weghe, N., & Verstockt, S. (2023). *Evaluating Segmentation Approaches on Digitized Herbarium Specimens*. In: Bebis, G., et al. Advances in Visual Computing. ISVC 2023. Lecture Notes in Computer Science, vol 14362. Springer, Cham.

2. **Milleville, K.**, Van Ackere, S., Van de Weghe, N., Verstockt, S., & De Maeyer, P. (2022). *An exploratory analysis on using social media to monitor environmental issues and natural disasters.* Proceedings of the 17th International Conference on Location Based Services (LBS 2022), 1–7.

3. Ali, D., **Milleville, K.**, & Verstockt, S. (2022). *NewspAIper: AI-Based Metadata Enrichment of Historical Newspaper Collections.* In DH Benelux 2022-ReMIX: Creation and alteration in DH (hybrid).

4. **Milleville, K.**, Thirukokaranam Chandrasekar, K. K., Blyau, T., Iannello, A., Michelucci, U., & Verstockt, S. (2022). *Extraction and classification of historical stamp cards using computer vision.* In DH Benelux 2022-ReMIX: Creation and alteration in DH (hybrid) (pp. 1-4).

5. **Milleville, K.**, Van den broeck, A., Vissers, R., Magnus, B., Vanderperren, N., Vergauwe, A., ... Verstockt, S. (2022). *FAME video browser – face recognition based metadata generation for performing art videos.* DH Benelux 2022: RE-MIX: Creation and Alteration in DH, Proceedings. Presented at the DH Benelux 2022 - ReMIX: Creation and alteration in DH (hybrid), Esch-sur-Alzette, Luxembourg.

6. Chandrasekar, K. K. T., **Milleville, K.**, & Verstockt, S. (2021). *Species Detection and Segmentation of Multi-specimen Historical Herbaria.* Biodiversity Information Science and Standards, 5, e74060.

7. Chambers, S., Lemmers, F., Pham, T. A., Birkholz, J. M., Ducatteeuw, V., Jacquet, A., Dillen, W., Ali, D., **Milleville, K**., & Verstockt, S. (2021). *Collections as Data: interdisciplinary experiments with KBR's digitised historical newspapers:*

a Belgian case study. In 7th DH Benelux: The Humanities in a Digital World (DH Benelux 2021).

8. **Milleville, K.**, Verstockt, S., & Van de Weghe, N. (2020). *Improving toponym recognition accuracy of historical topographic maps.* In International workshop on Automatic Vectorisation of Historical Maps. ELTE Eötvös Loránd University. Department of Cartography and Geoinformatics.

9. **Milleville, K.**, Ali, D., Porras-Bernardez, F., Verstockt, S., Van de Weghe, N., & Gartner, G. (2019). *WordCrowd: A location-based application to explore the city based on geo-social media and semantics.* In 15th International conference on Location Based Services (LBS 2019) (pp. 231-236). Vienna University of Technology. Research Group Cartography.

10. Vandecasteele, F., Kumar, K., **Milleville, K.**, & Verstockt, S. (2019). *Video Summarization And Video Highlight Selection Tools To Facilitate Fire Incident Management.* In ISCRAM.

11. Verstockt, S., **Milleville, K.**, Ali, D., Porras-Bernandez, F., Gartner, G., & Van de Weghe, N. (2019). *EURECA: EUropean Region Enrichment in City Archives and collections.* In 14th ICA conference: Digital approaches to cartographic heritage (pp. 161-169). Aristoteleio Panepistimio Thessalonikis (APTh).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Imagenet classification with deep convolutional neural networks*. Advances in neural information processing systems, 25, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *Imagenet: A large-scale hierarchical image database*. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[3] K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[4] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. *Dive into deep learning*. arXiv preprint arXiv:2106.11342, 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention is all you need*. Advances in neural information processing systems, 30, 2017.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv preprint arXiv:2010.11929, 2020.

[7] R. Bommasani et al. *On the Opportunities and Risks of Foundation Models*, 2022. arXiv:2108.07258.

[8] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun. *Alignedreid: Surpassing human-level performance in person re-identification*. arXiv preprint arXiv:1711.08184, 2017.

[9] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. *Faceforensics++: Learning to detect manipulated facial images*. In Proceedings of the IEEE/CVF international conference on computer vision, pages 1–11, 2019.

[10] B. M. Thiers, M. C. Tulig, and K. A. Watson. *Digitization of the new york botanical garden herbarium*. Brittonia, 68:324–333, 2016.

[11] *The Belgian historical gazetteer: a tool to map Belgian toponyms beyond linguistic and chronological borders*. Zenodo, June 2023. Available from: https://doi.org/10.5281/zenodo.7997174, doi:10.5281/zenodo.7997174.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. *Microsoft coco: Common objects in context*. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[13] Y. Zhao and Q. Zhu. *Evaluation on crowdsourcing research: Current status and future direction*. Information Systems Frontiers, 16(3):417–434, 2014.

[14] T. Causer and M. Terras. *Crowdsourcing Bentham: beyond the traditional boundaries of academic history*. International Journal of Humanities and Arts Computing, 8(1):46–64, 2014.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. *Segment anything*. arXiv preprint arXiv:2304.02643, 2023. Available from: https://arxiv.org/abs/2304.02643.

[16] OpenAI. *GPT-4 Technical Report*, 2023. arXiv:2303.08774.

[17] F. Gilardi, M. Alizadeh, and M. Kubli. *ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks*, 2023. arXiv:2303.15056.

[18] G. Jocher, A. Chaurasia, and J. Qiu. *YOLO by Ultralytics*, January 2023. Available from: https://github.com/ultralytics/ultralytics.

[19] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu. *Mobile-former: Bridging mobilenet and transformer*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5270–5279, 2022.

[20] C. Giattino, E. Mathieu, V. Samborska, and M. Roser. *Artificial Intelligence*. Our World in Data, 2023. https://ourworldindata.org/artificial-intelligence.

2

Photo Collection Enrichment

"A picture is worth a thousand words"

– Fred R. Barnard

This chapter discusses the automated processing of photo collections using computer vision. It starts with an overview of relevant techniques and related work. Next, it details the use of computer vision techniques to recognize objects, using both supervised and unsupervised methods. The chapter concludes with a facial recognition pipeline, that was successfully used to recognize over 62 thousand faces.

This chapter features an adapted version of the following publication:

Milleville, K., Broeck, A. V. D., Vanderperren, N., Vissers, R., Priem, M., Van de Weghe, N., & Verstockt, S. (2023). **Enriching Image Archives via Facial Recognition.** ACM Journal on Computing and Cultural Heritage. https://doi.org/10.1145/3606704

2.1 Introduction

In recent years, cultural heritage and archive institutions have digitized their photo collections. This digitization process entailed many challenges, which are discussed in Sections 1.1 and 1.3. Often, only a generic image description and date are

available. The names of the persons depicted might not be available in the image metadata or this metadata might not be complete (e.g. "Group photo of the Flemish Parliament, 1984"). This makes it difficult to search for specific persons, objects, or places of interest in the entire collection.

As collections can contain millions of images, manually annotating each image with detailed metadata is not a feasible task. Therefore, manual annotation efforts often focus on a smaller part of the collection that provides the most historical value. Crowdsourcing approaches can also help speed up the annotation efforts [1–3]. These require a large enough set of volunteers willing to help annotate the collections and some costs to set up the crowdsourcing infrastructure and marketing. The resulting metadata still needs to be validated in some way before adding it to the collections, as it can be difficult for people not familiar with the source material to provide accurate labels. Automated metadata creation techniques using AI have been gaining popularity, as the quality of the techniques has improved dramatically over the years. While this was not a focus of the performed research, many challenges remain concerning bias and fairness in AI and how to better integrate these results in archival information systems [4]. For instance, multiple studies have shown that facial recognition systems tend to have a higher error rate for certain demographics and even non-demographic attributes like eyeglasses and accessories [5, 6]. Despite these issues, computer vision models are an effective tool to analyze large picture collections. They are robust, efficient, and generalize well to similar datasets.

2.2 Recognizing Objects of Interest

To find objects or places of interest in a given image, there are typically four main approaches. These are classification, object detection, object segmentation, and image retrieval approaches.

2.2.1 Image Classification

Image classification models will assign one or more object classes to a given image, without any information on where this object is located in the image. This was the main task of the ImageNet challenge [7], where the goal was to predict a single class for the test images of the dataset. The ImageNet dataset contains over one million labeled training images and features 1000 object classes. Because this dataset is so diverse, it is still frequently used to pre-train computer vision models. People noticed that CNNs trained on ImageNet can be reused on other datasets with little modifications, by replacing the final classification lay-

ers. This practice is referred to as transfer learning. It enables neural networks to achieve good results on a new dataset, with much fewer labeled images and with a faster convergence. The lower layers of the network have essentially become generic image feature extractors.

Image classification approaches are very efficient and used when it does not matter where the object is located. One common example is using them to detect inappropriate or not safe for work (NSFW) content. Many social media platforms employ such models, where any image that is predicted as NSFW is either immediately removed or flagged for manual validation. Image classification could also be used to separate different types of images as part of an ensemble method, where each type of image is then fed to a second, specialized model. For instance, a classification model can detect if buildings are prominently visible (see Figure 2.1). If a building was found, the image can be fed to a secondary model to extract building-specific keypoints (e.g. via [8]). Classification is also the easiest and fastest approach to label. While predicting one or more object classes for a given image can be useful, it does not give any information on the number or location of these objects.

Figure 2.1: Image classification can be used to differentiate photographs of buildings (left) versus people (right).

2.2.2 Object Detection and Segmentation

To locate all visible objects in a given image, object detection can be used. Such an approach will predict a bounding box location and class for each recognized object. Depending on the model, these bounding boxes can be rotated or not (see Figure 2.3 for an example of non-rotated bounding boxes). They are represented

by two or four coordinates. Detection models typically contain additional regression layers, which take input features and produce the coordinates of the bounding boxes [9]. However, when the visible objects have complex or non-convex shapes (e.g. with holes or consisting of multiple parts), these bounding box predictions will not be an accurate representation of the object's shape.

Therefore, image segmentation techniques can be used to predict pixel masks for each recognized object or object class. Segmentation techniques are categorized into three tasks: semantic, instance, and panoptic segmentation [10]. Semantic segmentation aims to classify each pixel in the image (e.g. "person", "car" or "background"). So, every pixel representing a person on the image is assigned the same class. Instance segmentation not only classifies each pixel but distinguishes between individual instances of each class. Panoptic segmentation combines these two types, segmenting both "stuff" (semantic) and "things" (instance) within the same framework. Figure 2.2 visualizes the three segmentation types.

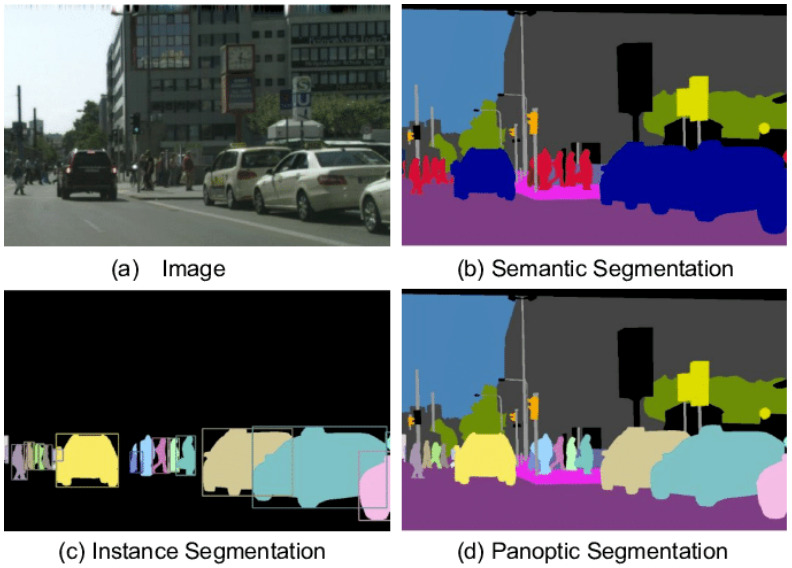


Figure 2.2: Different segmentation types visualized (Image from [10])

While segmentation is more accurate, the masks are much harder and time-consuming to label than bounding boxes. The segmentation labels are typically represented as a polygon or binary mask. One of the most popular datasets containing both detection and segmentation masks is COCO [11]. This dataset contains over 200 thousand labeled images, featuring more than 1.5 million annotations. Detection models trained on such datasets are already widely applicable out-of-the-box and can easily be fine-tuned via transfer learning. Popular detection

models, such as YOLOv8 [12] and Detectron2 [13], achieve accurate results, while still being performant and can easily be integrated into existing workflows. Figure 2.3 shows the output of a YOLOv8 instance segmentation model on an archive photograph, with both the masks and bounding boxes visualized. It is clear that the segmentation masks better represent the recognized objects.

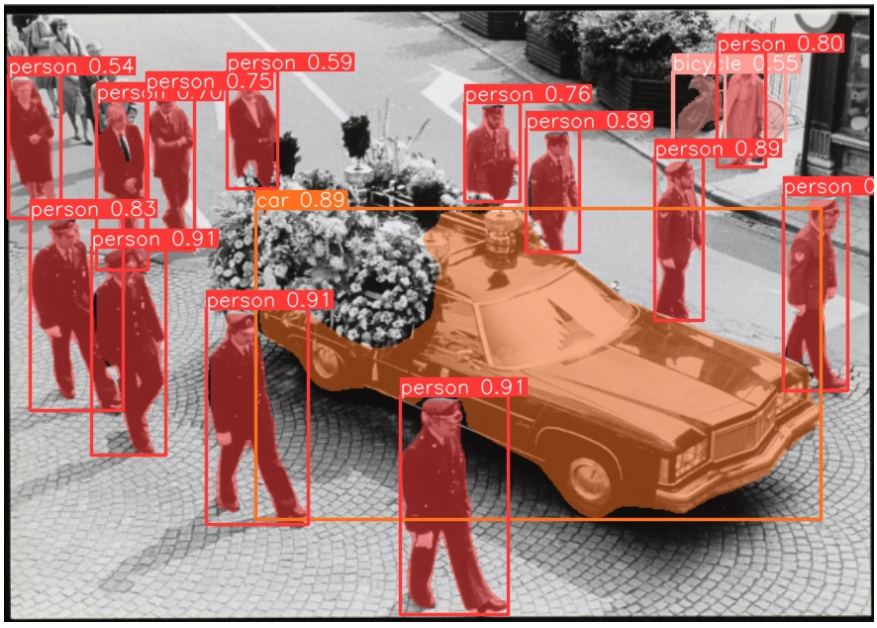


Figure 2.3: Object instance segmentation using YOLOv8 [12]. The object class and confidence score are shown above each detection.

To conclude, pre-trained object classification, detection, and segmentation models are readily available, with a wide range of object classes. For most pictures, these can be used as is, without any modification. When the goal is to find a specific object or building, detection models will need to be fine-tuned using additional labeled data. This may not be optimal, especially if the number of distinct buildings or objects is quite large.

2.2.3 Image Retrieval

Imagine you have a dataset of archive pictures, taken across an entire city or country. If you now want to determine where these pictures were taken, an object detection approach will not get you far. You would have to label distinct landmarks or buildings, that are commonly featured in these pictures. This can quickly get out

of hand and become counterproductive. Therefore, an image retrieval approach is often preferred for such use cases.

Image retrieval approaches will try to find the top-k most similar results from a dataset, for a given query image. They generally use image similarity models, that predict a similarity score for a pair of images. During training, an image similarity model is typically trained on pairs of images, containing both correct and incorrect matches for each person or object. The model's loss function is constructed in a way that maximizes a similarity metric between learned embeddings from image pairs containing the same object and minimizes it otherwise. To use such a model in practice, you could feed it image pairs, but such an approach is not scalable. If you have a dataset of 10,000 pictures and want to find the most similar pictures for a couple of query images, the model would have to predict this score 10,000 times for each query image. Therefore, image similarity models typically remove the last layers of the network during inference, resulting in an output vector. This vector or embedding, is a low-dimensional representation of the input image. To match two embeddings, a similarity metric is used. Typical examples of such metrics are Euclidean distance and cosine similarity.

So, if we want to estimate where a picture was taken given our archive picture collection, we first extract an embedding for each image in the dataset. Then, we extract an embedding for the new pictures and use a similarity metric to find the top-k most similar matches. The big difference in this approach is that the embeddings only need to be calculated once for each image of the dataset.

Image similarity approaches are commonly used for landmark identification and face recognition models [14, 15]. We successfully used such an approach in the Ugesco [16] project, to automatically geolocate a dataset of archive pictures taken in Brussels during the Second World War. We constructed a small set of pictures featuring popular locations and landmarks from Google images and Google Streetview. Then, we extracted image embeddings for each of them and calculated the similarity scores with the archive dataset. This resulted in an ordered list of matches for each image. The proposed image matches were then subsequently validated via crowdsourcing. Figure 2.4 shows the resulting matches for the Ugesco dataset, given two recent images. Because we know the location of our recent images, we can assign this location to the archive images, automatically geolocating them.

2.2.3.1 Multimodal Models

Besides images, similarity and retrieval approaches can also be used on text, which became increasingly popular after the Word2Vec [17] model was released. They constructed embeddings from a given string to match it with other text. This resulted in a semantic similarity between two texts, instead of a simple string sim-

Figure 2.4: Two recent query images, taken from Google, and their best-matching results from the Ugesco dataset.

ilarity. In 2021, the popular CLIP (Contrastive Language–Image Pre-training) [18] model was published. This model combined image and text embeddings by pre-training 400 million image and caption pairs. The model was trained to predict the caption for a given image. Such an approach made it possible to match images with text and vice versa. It also showed impressive zero-shot accuracy on ImageNet.

Many future research used a similar approach to CLIP, to combine text and images into a single, unified model. One such multimodal model is BLIP-2 [19], which we used in a case study on the "Collectie van de Gentenaar"[1] (CoGent). This is an open dataset, containing pictures with descriptions and tags from multiple museums and institutions from Ghent.

First, we extracted an embedding for each image in the dataset. Next, we calculated the pairwise similarity scores, resulting in an ordered list of matches for each image. Given a new image, the embedding can be calculated and quickly matched with the dataset using an efficient vector similarity approach such as Annoy[2]. Because many images lack a good description, these can be difficult to find, therefore image similarity offers an additional way of querying the database. Figure 2.5 shows three examples of image-to-image matching results and their cosine similarity scores.

Besides image-to-image matching, we also wanted to find images related to typical Belgian topics such as cycling, chocolate, etc. We constructed a list of topics and associated keywords. Then, we used BLIP-2 to embed these texts and find the best-matching images from the dataset, using the image embeddings we already calculated. This way, we have found the best-matching images for each topic. Text embeddings enable users to search the visual content of the collection, using natural language. The BLIP-2 library also features an image captioning model, which can generate image descriptions. While these descriptions are rather generic,

[1] https://data.collectie.gent/

[2] https://github.com/spotify/annoy

Figure 2.5: Best matches from the CoGent dataset for three query images.

they can be directly integrated with existing database search methods. Figure 2.6 shows three sample images from the dataset, with their auto-generated captions. We noticed that the model sometimes gave incorrect captions, so its results should be used with caution.

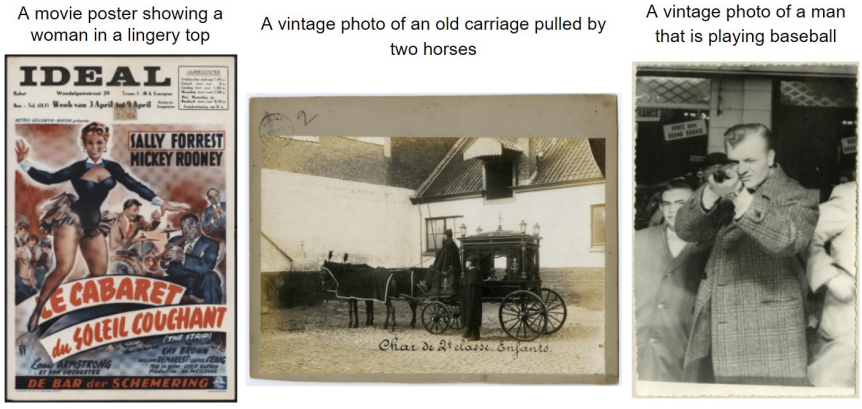


Figure 2.6: Example of some auto-generated image captions. The man pictured on the right is definitely not playing baseball.

While not perfect, the CoGent use case demonstrates the added value of using multimodal similarity-based methods to enrich the collection. The presented methods required no manual annotation, except for a list of topics and keywords. They allow users to search for visual content using natural language or similar

images. The methods do require some processing, to extract embeddings from a new search query (a query image or natural language). This is their main limitation and the biggest reason why such approaches aren't already integrated in many collections.

2.3 Recognizing Persons of Interest

To efficiently and accurately find persons of interest in large archive collections, we developed a generic image enrichment pipeline based on state-of-the-art facial recognition tools. This pipeline was successfully used in the FAME (facial recognition as a tool for metadata creation) project to enrich multiple image archive collections. FAME was a collaborative project with Meemoo (the Flemish Institute for Archives) and four content partners: Kunstenpunt (the Flanders Arts Institute), KOERS (the Museum of Cycle Racing), ADVN (Archive for National Movements), and the Flemish Parliament Archive. Each content partner provided a sample dataset to enrich and assisted in the validation process. The pipeline enriches the collections with image-level metadata, by predicting who is depicted in each picture. It was applied to a dataset containing over 150 thousand images, resulting in more than 62 thousand confident person predictions. Furthermore, an interactive labeling tool was developed to validate the person predictions. Using this tool, more than 180 thousand annotations were collected over one year and used to validate the recognition model accuracy. This additional metadata is a valuable resource for researchers and others interested in exploring and interpreting the contents of the image archives. Researchers can more easily analyze the trends and relationships of the identified persons throughout the collection. Facial recognition can also aid in the identification of previously overlooked individuals or groups, providing a more comprehensive understanding of historical events.

2.3.1 An overview of Facial Recognition Systems

Facial recognition has been a prominent area of computer vision research for quite some time. It tackles the problem of assigning the correct identity to a given face, using a dataset of known faces, also called the reference or ground truth set. Generally, a face recognition model consists of three components. First, a face detector detects and localizes the faces present in the image. Research about face detection models has been going on since the 1990s, with the earlier works using low-dimensional handcrafted features [20]. At the time, such approaches were state-of-the-art but were later vastly outperformed by convolutional neural

networks (CNNs), as were most other computer vision problems. Such CNNs are similar to those used for object detection. They generally consist of a backbone (feature extractor) followed by some additional layers, that perform bounding box regression to predict the location of each face in the image. One of the more popular face detection models, RetinaFace, is very accurate, while still having a relatively fast processing time [21]. Other popular face detection models include Mediapipe [22] and MTCNN [23].

Next, an alignment module crops and normalizes the detected faces. This module preprocesses the detected faces for the final component to perform face recognition. This final component extracts face embeddings (feature vectors) from a given face, which can then be matched with a dataset of known faces (and extracted embeddings) [24]. To match two embeddings, a similarity metric is used. To recognize a known person from a given face, this metric is calculated for each embedding in the reference set. One can subsequently get the best matching person from the reference set and its similarity score. If this score is higher than a predefined threshold, it is considered a correct match.

Deepface [25], was the first face recognition approach to rival human performance and a big breakthrough in the field. They used a nine-layer CNN and achieved an accuracy of 97.35% on the LFW dataset [26]. Later, FaceNet [27] improved the accuracy to 99.63% using a retrained GoogLeNet on a private dataset and a triplet loss function. Since then, many new open datasets arose, such as VGGFace2 [28], CASIA-Webface [29], and WebFace260M [30], which led to multiple advances in the field of face recognition. For our facial recognition pipeline, we used InsightFace, which is one of the most popular open-source face recognition libraries. They offer close to state-of-the-art accuracy and great performance with their pre-trained models [14, 31]. For a more detailed research overview on face recognition, we refer the reader to [24].

2.3.2 Facial Recognition in Practice

In [32] the location and geometry of facial landmarks were used to detect the facial expressions and their intensity for Indian arts performers. The automated system demonstrated an accuracy exceeding 95% on a varied dataset of performers, with and without makeup. Via user feedback, they learned that both beginners and experts found the automated system useful to practice and improve their expressions during performances. [33] proposed a three-part methodology for analyzing gender in historical advertisements. The authors covered face and gender detection as well as the detection of visual medium types (illustrations or photographs). From their larger collection of digitized newspapers, they annotated a sample of 45 thousand images containing both photographs and illustrations with the location and gender of visible faces. This resulted in a dataset of nearly

19 thousand annotations, on which they benchmarked a pre-trained DSFD [34] and RetinaFace model. While the models' performance suffered when processing newspaper images rather than photographs, they still achieved an average precision of 0.71 and 0.68, respectively. They subsequently retrained a classifier and were able to accurately distinguish male and female illustrations or photographs, with an average weighted precision of 0.84. In [35], the authors used a facial recognition pipeline to automatically recognize celebrities and prominent persons from the French national video archives. They constructed their reference set of known persons via web scraping. For each person, they gathered 50 images from the web and filtered out wrongly detected faces using the extracted face embeddings. The predictions were validated on two samples from their dataset. One consisted of 216 shots and 13 persons of interest, the other contained 100 shots featuring six persons of interest. They achieved a recall of 0.97 and 0.98, respectively. Photo Sleuth [1] is a web-based platform that combines facial recognition with crowdsourcing to identify Civil War portraits. Due to their age and quality, these Civil War pictures were naturally more difficult to process accurately than modern digital photographs. Therefore, they used additional visual information (coat color, chevrons, shoulder straps, etc.) and help from the crowd to improve the recognition process. In a one-year period, over 12 thousand users registered to the platform. They helped identify 2979 portraits and uploaded over 8000 new images to the platform, demonstrating the added value of crowdsourcing.

2.3.3 Overview of the Pipeline

An overview of the entire facial recognition pipeline is given in Figure 2.7. The first step in the pipeline is to build a reference set of known persons we aim to recognize. This process was performed semi-automatically, with manual validation. Face detection and feature extraction were performed on each of the reference images. Next, face detection and feature extraction were performed on the unlabeled datasets. After this extraction, each detected face was matched with all the faces in our reference set via a similarity metric on the embeddings. This resulted in a person prediction and associated score for each detected face. The faces that did not have a confident prediction were clustered, to find frequently occurring persons that were not included in the reference set. The final step of the pipeline was to manually validate these person predictions and found clusters via an interactive labeling tool. The resulting metadata was then exported back to the content partners.

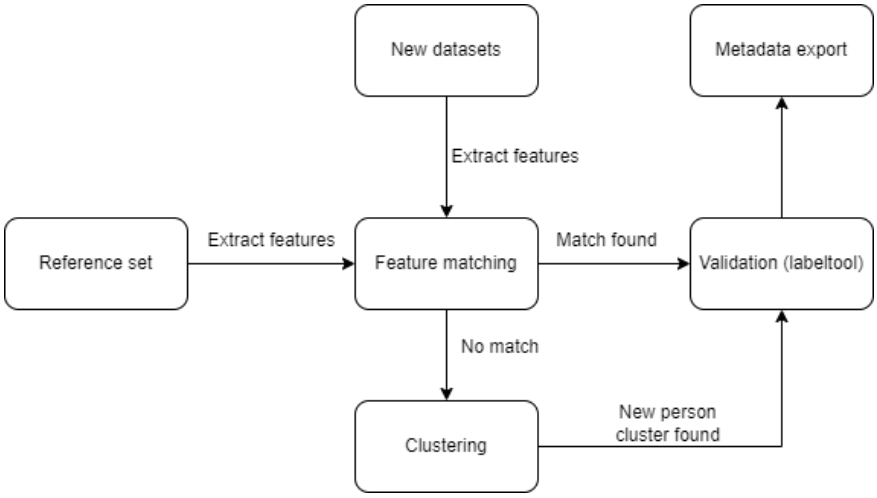


Figure 2.7: Overview of the facial recognition pipeline.

2.3.4 Building the Reference Set

To perform facial recognition, we first need a reference set of known persons and their associated embeddings. Ideally, each person should have a set of images that only contain the person in question. Having multiple reference images is preferred, to increase the robustness of facial recognition. These should be relatively high-quality images, with the person facing the camera. To build this reference set, a list of persons of interest for each content partner was made using expert knowledge, image descriptions, and Wikidata. Images for each person were collected via web scraping, Wikidata, and manually selected from the archives of Meemoo and our content partners using the associated metadata.

We split our reference dataset into three collections based on the type of source material and persons of interest. The first collection (Kunstenpunt) contains performing artists, actors, and other people from the cultural sector. The second collection (Koers) consists mainly of cyclists and images taken during cycling events. The third collection (Government) consists of politicians and well-known activists. Face detection and feature extraction were performed on each of the images of our reference set, using the pre-trained model pipeline from InsightFace (buffalo_l [3]). This pipeline consists of several models, including an SCRFD [36] model for face detection and a Resnet50 model with Arcface loss trained on WebFace600K for face recognition, which generate the following outputs:

[3] https://github.com/deepinsight/insightface/tree/master/python-package#model-zoo

- Location and detection confidence for each detected face in the image
- Facial keypoints and landmarks (both 2D and 3D) for each detected face
- A 512-dimensional feature vector (embedding) for each detected face
- Age and gender prediction for each detected face

These outputs were subsequently grouped per person in the reference set, resulting in a set of embeddings for each person. Some of the images in the reference set had problems regarding the face detection. If no faces were detected, which sometimes occurred if the face was large w.r.t. the image (e.g. close-ups), the input image size of the model was lowered from the standard 640x640 pixels to 256x256 pixels. This often led to a successful detection. If multiple faces were detected, mainly due to people in the background or false detections, it is difficult to be certain which face belongs to the person in question. Filtering the wrong detections based on their relative size and position frequently led to errors. We solved most of these errors by matching each detected face with all the other faces from that person in the reference set. By using the resulting similarity scores, it was trivial to decide which face (if any) belonged to the person in question. If the image still had problems (no face detected or not sure which face was correct), it was discarded from the reference set.

Ultimately, we ended up with 60,976 images portraying 6075 unique persons across the entire reference set. Table 2.1 gives an overview of each of the collections in our reference set. It features the number of images and unique persons, the average number of images per person, and the number of images that were discarded. The Koers collection had the most images with problems, mainly due to the gathered images containing multiple persons or the person's face not being clearly visible (many pictures were taken during a cycling race).

Table 2.1: Overview of the reference set

Collection	Images	Unique persons	Images/person	Discarded
Kunstenpunt	37,172	2393	15.53	92 (0.25%)
Koers	15,323	2791	5.49	482 (3.05%)
Government	8481	891	9.51	7 (0.08%)
Total	60,976	6075	10.04	581 (0.94%)

2.3.5 Face Recognition on the Archive Data

After constructing the reference set and extracting a set of embeddings for each person, we can use these to perform face recognition on the unlabeled archive images. Similar to the reference set, this dataset was split into three corresponding collections, depending on the content partner. For Kunstenpunt, a sample of 17,382 images from their digitized and born-digital archive was used (1933-2020). For Koers, we used a set of 123,911 digitized photographs taken by Maurice Terryn that cover cycling events between 1969-1980. For the Government dataset, we used 5587 pictures from ADVN and 4824 from the Flemish Parliament Archive (1975-2019). The same InsightFace model pipeline was used to detect the faces on each image and extract their embeddings.

To predict the persons present in the images, the detected faces and corresponding features need to be matched with our reference set features via a similarity or distance metric. Euclidean distance on the normalized features is often used, but we chose to use cosine similarity, as initial testing on the reference set (matching faces of the same person) showed it was slightly more accurate. The cosine similarity is given by Eq. 2.1 and is bounded by [-1, 1], where 1 indicates a perfect match (same embeddings).

$$\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

For each collection, the cosine similarity between the detected faces' embeddings and all the embeddings in the reference set was calculated. This resulted in a list of similarity scores for each detected face from the unlabeled dataset. From this list, the 100 highest scores and associated person names (from the reference set) were saved. As a person in the reference set usually has multiple images and embeddings, the correct person prediction will likely appear multiple times in this list. Figure 2.8 visualizes the similarity scores and top person predictions for a sample image of the ADVN archive collection. Both persons were accurately recognized, with similarity scores of 0.66 and 0.75. Because of the large number of reference set embeddings, these calculations can become memory and time-consuming. Therefore, an efficient approximation approach such as Annoy can be used to speed up this process. We used such an approximation for a part of the evaluation (see 2.3.9).

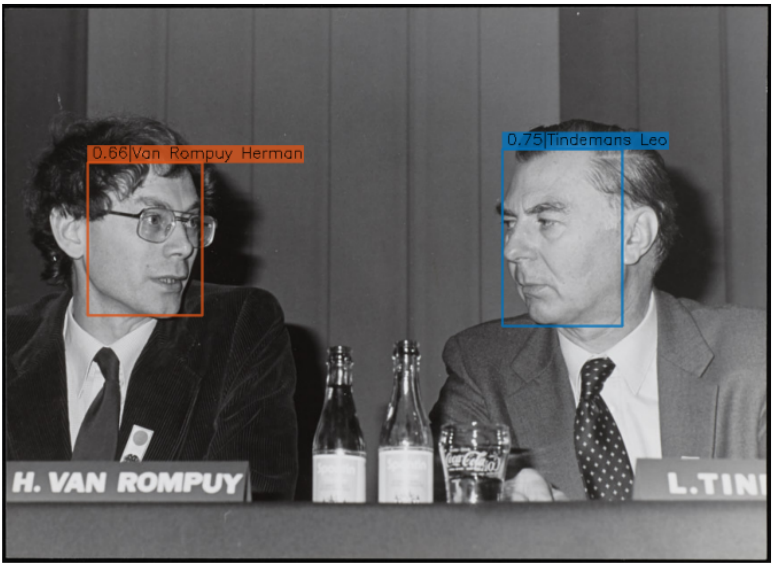


Figure 2.8: Sample image from the ADVN collection depicting a young Herman van Rompuy (left) and Leo Tindemans (right). The correct person predictions and similarity scores (0.66 and 0.75, respectively) are visualized.

2.3.6 Finding New Persons

After performing the face recognition on the unlabeled datasets, a large number of detected faces may have low similarity scores and poor person predictions. This is often due to the detected faces being too small or noisy, resulting in bad embeddings. When this is not the case, the poor prediction is usually because the correct person is not included in the reference set. However, this person may appear multiple times in the unlabeled dataset. Therefore, we can use a clustering approach on all detected faces with a poor prediction, to find persons not included in the reference set.

First, the detected faces were filtered on their size. Detected faces with a width or height smaller than 80 pixels were ignored. This removed many false positive detections and smaller faces of people in the background. Next, only faces with a top similarity of less than 0.5 were considered. The cutoff value of 0.5 was chosen as it usually resulted in a correct prediction (see 2.3.8.2). For each collection, an initial test was done using the DBSCAN clustering algorithm, which clusters higher-density regions [37]. Because DBSCAN expects a distance metric and not a similarity metric, we used the inverse of the absolute value of the previously calculated cosine similarity. This distance metric ranges from $[0, 1]$, where 0

indicates a perfect match. We set the epsilon parameter to 0.5 and the minimum number of samples per cluster to 5. This ensures that the faces found in each cluster will be relatively similar to each other. Figure 2.9 shows some sample images from two found clusters from the Kunstenpunt collection. Both persons were not included in the initial reference set but still appeared multiple times in the Kunstenpunt archive collection. After manual validation, these persons can now easily be added to the reference set.

Figure 2.9: Sample images of two clusters found for persons not included in the initial Kunstenpunt reference set.

2.3.7 Applying the Model to Video

The content partners also provided a small sample of videos to perform a feasibility study on. As a video is just an ordered set of frames, we could perform face recognition on every frame. However, such an approach would not be scalable and many processed frames would result in the same prediction. If a person is in frame for one minute and the video has a frame rate of 30 fps (frames per second), he would appear on 1800 consecutive frames. The number of frames to process can be reduced by skipping a fixed number of frames after one has been processed, but this will still result in a large number of frames. Instead, a test was performed using a scene detection algorithm [38], that detects when there's a different scene or shot in the video. For each scene, we randomly selected three frames, which were subsequently processed for face recognition. Because the scene detection algorithm runs much faster than face recognition, this approach was more efficient. For instance, one video of the Flemish Parliament with a frame rate of 25 fps and a duration of 1h44 minutes contained a total of

156,000 frames. However, using scene detection, 85 scenes were detected and a total of 255 frames were extracted for face recognition. This approach works well when the video has many cuts or major shot transitions, as these were usually detected by the algorithm. Virtual meetings are a great example of such type of videos, as the camera switches to the active speaker. For longer shots with a static camera, where persons are moving in and out of the shot, the scenes were often quite long and their detection less accurate. Using this approach, the resulting person predictions were grouped per detected scene, providing fine-grained metadata. We used this metadata to develop a video browser web application[4] that allows the user to query the video based on the recognized persons and find all the shots they appeared in [39]. Figure 2.10 shows the video browser interface for a video meeting of the Flemish Parliament. The different scenes are highlighted and grouped per recognized person, allowing for quick navigation of the video.

2.3.8 Evaluation

This section details the interactive labeling tool, which was used to validate the face recognition predictions. It also details the evaluation procedure and presents the face recognition accuracy. Furthermore, an exploratory study was performed analyzing the detected faces and predicted genders. The section concludes by comparing the InsightFace embeddings with FaceNet, another popular face recognition model.

2.3.8.1 Validating the Model Predictions

Manually validating each person prediction is a very time-consuming process, considering the size of our dataset and the number of persons in the reference set. Each image can contain multiple persons, further increasing the time spent labeling the predictions for each image. Therefore, an interactive web-based labeling tool was developed, inspired by our earlier work [40]. Instead of labeling each image or prediction one by one, the predictions are grouped on a per-person basis and presented in batches for labeling. After selecting a person, some sample images of the reference set are shown, together with a batch of faces for which the selected person was predicted. Figure 2.11 shows this interface after selecting the person 'Bert Anciaux' from the Government collection. The predictions are sorted by their similarity score, with the most confident predictions being shown first. Above each face, metadata of the original image is given, and additional metadata is shown on right-click.

[4] https://labeltool.idlab.ugent.be/eureca/label/faces/videobrowser/index

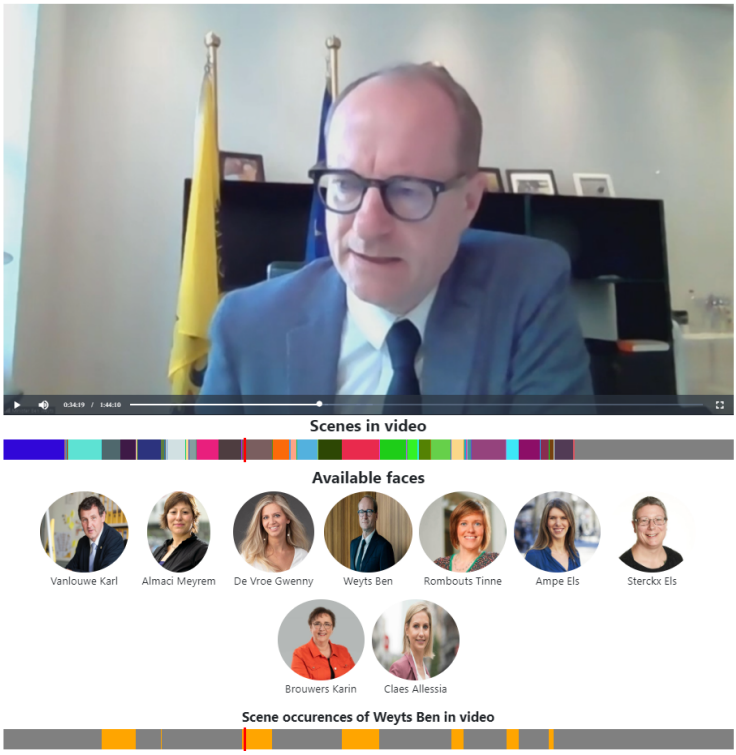


Figure 2.10: Interface of the video browser application for a virtual meeting of the Flemish Parliament. The recognized persons are shown below the video and their scene occurrences are visualized after clicking on their portrait.

The tool allows the user to quickly select multiple predictions and label them with the same value. We opted for three possible labels: accept, reject, and bad. A prediction was accepted if the selected face belonged to the selected person and rejected otherwise. The bad label was used to indicate that the selected face was either a false detection (no face visible in the crop) or it was too noisy to recognize the person. This can happen for small faces in the background, or when the face is viewed from the side. Some examples of such bad faces are shown in Figure 2.12. Without any knowledge of the context, it is extremely difficult to determine the correct person. Luckily, the similarity scores for such bad faces are much lower on average.

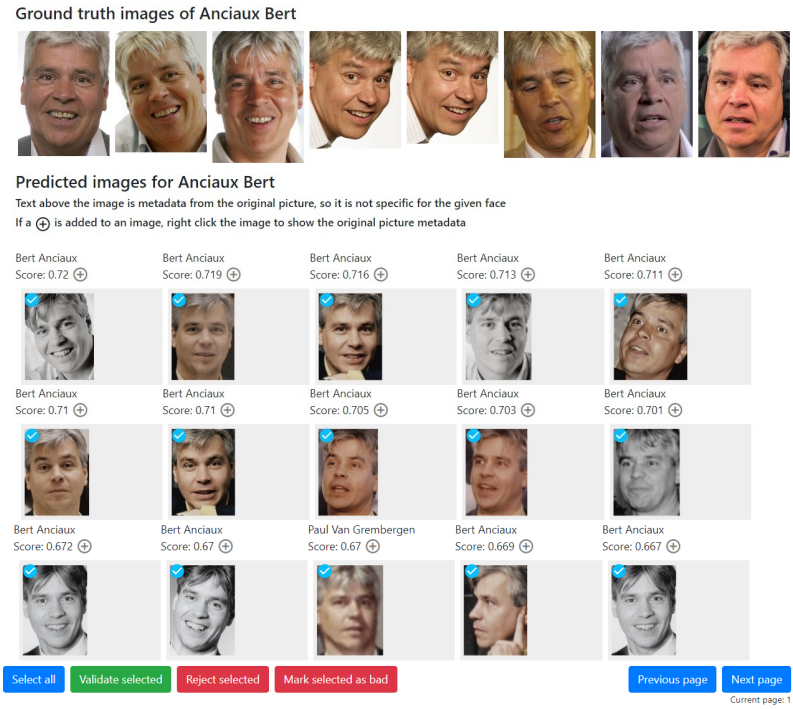


Figure 2.11: Interface of the labeling tool, after selecting a person (Bert Anciaux). The top row shows reference set images of that person, with the top predicted faces below it. The metadata of the original image is shown on top of each predicted face. The user can validate multiple predictions at once, speeding up the labeling process.

2.3.8.2 Results

Over a one-year period and in collaboration with the content partners, we have collected a total of 182,202 face prediction labels and 2053 labels for face clustering via the labeling tool. For the Koers collection, the initial dataset consisted of a small sample of 6051 images, which was later expanded with an additional 117,860 images. More than 1 million faces were detected on this additional dataset. Therefore, the person predictions were first filtered on a minimum similarity score of 0.4, to focus labeling efforts on more confident predictions. Table 2.2 breaks the validation labels down by collection and label value. Roughly 43.4% of the face predictions were accepted, 44.4% rejected, and 12.2% labeled as bad faces. Because the person predictions were sorted by similarity score, these labels are biased towards acceptance, as not every detected face in the dataset was labeled. The average similarity score for predictions with an accepted label was 0.57, for a

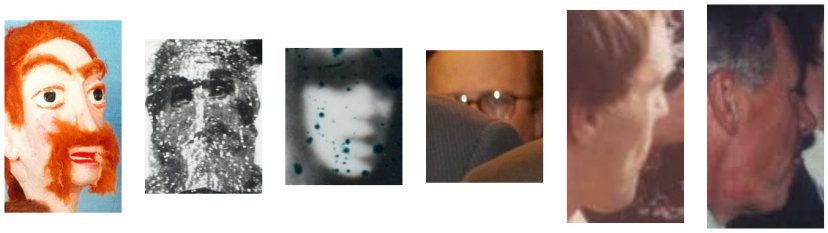


Figure 2.12: Examples of 'bad' faces detected by the face detection model. These include masks wrongly detected as a face, faces that are too noisy to tell who is depicted, and obstructed faces.

rejected label 0.31, and for a bad label 0.29. The full similarity score distributions of all labeled predictions are shown in Figure 2.13. There is a clear spike visible at 0.4, due to the filtering of the Koers dataset. Nevertheless, these distributions clearly show that a higher similarity score increases the probability of a correct person prediction.

Clearly, there are correct predictions with lower scores and incorrect predictions with higher scores. A minimum score threshold can be set, to mark confident predictions as correct and discard the others. A higher threshold will result in fewer errors, but also fewer correct person predictions with higher scores. This is the classic trade-off between precision and recall. Table 2.3 lists the various precision and recall scores for thresholds between 0.4 and 0.8. The threshold of 0.5 resulted in a high precision of 0.936 and still had a relatively high recall at 0.740. 62,455 person predictions had a similarity score that was greater than or equal to this threshold. Meanwhile, the highest threshold of 0.8 had an almost perfect precision of 0.997 (only 2 errors made) but the recall dropped to 0.01. Therefore only 1% of the correct predictions would remain after applying the threshold.

Table 2.2: Overview of the manually validated labels per collection. The number of 'bad' labels is much lower for Koers, due to the filtering on a minimum similarity of 0.4.

Collection	Labels	Accepted	Rejected	Bad
Koers	131,909	63,967 (48.5%)	63,558 (48.2%)	4384 (3.3%)
Government	42,040	12,245 (29.1%)	15,430 (36.7%)	14,365 (34.2%)
Kunstenpunt	8253	2818 (34.1%)	1939 (23.5%)	3496 (42.4%)
Total	182,202	79,030 (43.4%)	80,927 (44.4%)	22,245 (12.2%)

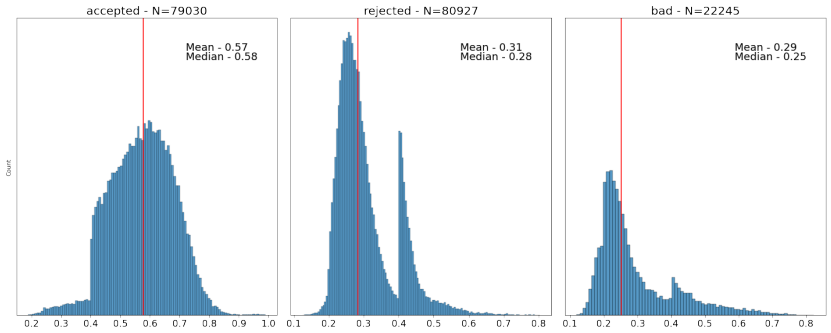


Figure 2.13: Distribution of person prediction similarity scores for the manually validated predictions, grouped by their label (accepted, rejected, and bad). The red lines indicate the median score. There is a clear spike visible at 0.4, due to the filtering of the Koers dataset.

2.3.9 Analysis

This section details a number of analyses made on the collection, using facial recognition and gender predictions. Furthermore, we show how person co-occurrences can be used to find connections in the dataset.

2.3.9.1 Number of Persons per Image

The number of detected faces per image can be a useful metric for various purposes. It can provide insights into past events and improve the search capabilities and accessibility of the collection. For example, it enables the search for large crowd gatherings or single-person profile photographs. For each collection, we filtered out the images on which no faces were detected. Next, the images were categorized based on the number of detected faces. Images featuring a single person, 2-4 persons, 5-10 persons, and more than 10 persons. For each category, the number of images was aggregated and divided by the total number of images for that collection. This process was performed for all detected faces and also for detected faces that were matched with a minimum score of 0.5. The resulting ratios are visualized in Figure 2.14. Most images feature a single person, except for the Kunstenpunt collection. There, the ratio for 2-4 persons is slightly higher because most of the images were taken during a performance, where multiple people are visible. The Government collection features the most images with more than 10 detected persons, due to a large number of group pictures and pictures taken inside the Flemish Parliament. After filtering the detected faces based on their person prediction score, we see a much lower number of persons per image.

Table 2.3: Precision and recall per similarity score threshold. The positive prediction column denotes the number of predictions with a similarity score greater than or equal to the threshold.

Threshold	Positive predictions	Precision	Recall	F1
0.40	97,716	0.782	0.967	0.865
0.45	75,978	0.901	0.867	0.884
0.50	62,455	0.936	0.740	0.826
0.55	48,839	0.957	0.591	0.731
0.60	34,447	0.971	0.423	0.589
0.65	20,727	0.978	0.256	0.406
0.70	9612	0.985	0.120	0.214
0.75	3151	0.991	0.040	0.076
0.80	768	0.997	0.010	0.019

The reason for this is twofold: many persons present in the collections were not included in the reference sets, and smaller faces in the background will generally have a lower prediction score. Nevertheless, 84 group photos of the Government collection feature more than 10 confidently recognized persons.

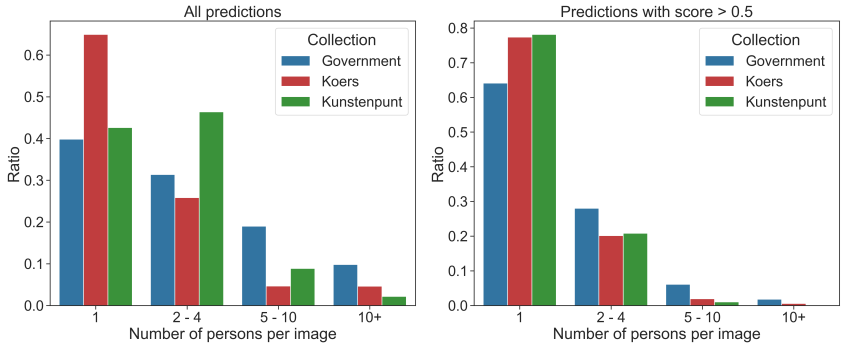


Figure 2.14: Distributions visualizing the number of detected faces per image for all three collections, without filtering (left) and after filtering on a minimum prediction score of 0.5 (right).

2.3.9.2 Network Analysis

In addition to aggregating the number of person detections, we identified connections between individuals using predicted person co-occurrences. For every image, predictions were filtered on a minimum similarity score of 0.5. Next, a graph was constructed with all the predicted person names as the nodes. Each time two individuals appeared in the same image, an edge between them was added to the graph, or the weight of the existing edge was incremented. Using this graph, we can calculate various metrics like degree centrality, to find the most connected persons in the collection. The degree centrality of a node is equal to the fraction of nodes it is connected to. We have performed this analysis on the collection of Koers. We compared the total number of occurrences of the 15 most frequently occurring persons with their degree centrality. Table 2.4 lists these persons with their degree centrality and Figure 2.15 visualizes the resulting network graph for these persons (in red) and their connections. As the total number of unique persons is over 1000, the full network graph becomes complex to visualize, therefore we limited the visualization to a subgraph of the 50 most frequently occurring persons.

If we compare the number of occurrences and degree centrality, we see that these have a strong correlation. However, some persons have a much lower centrality compared to others in this list. For instance, De Vlaeminck Erik has a high number of occurrences but a low centrality, indicating that he is featured with a relatively small fraction of the total number of unique persons identified. On the other hand, Demeyer Marc has fewer total occurrences but is pictured with more than double the number of unique persons. Next, we looked at the most frequently occurring duos. The most frequently occurring duo was Pollentier Michel and Maertens Freddy at 170 occurrences, followed by Merckx Eddy and Sercu Patrick at 149 occurrences. These duos were teammates for a portion of the Koers collection period, so they were often pictured together during or after a race.

2.3.9.3 Gender Prediction

Besides the number of detected persons per image and their connections, a small study was performed on the predicted gender of a subset of the collection of the Flemish Parliament (1980-2019). We chose to limit this study to this collection, as every image in this collection was accurately dated, which was not the case for the other collections. Also, the collection of Koers would be less interesting to study, as it features images taken during men's cycling races, which would heavily skew the results. The InsightFace model outputs a binary gender prediction for each detected face. This binary classification is a common practice in facial recognition due to its simplicity and accuracy. We acknowledge the limitations of

Table 2.4: Top-15 most frequently identified persons in the Koers collection, with their associated degree centrality.

Name	Occurrence	Centrality
Maertens Freddy	2632	0.210
Merckx Eddy	2019	0.179
Sercu Patrick	2007	0.159
De Vlaeminck Roger	1720	0.148
Godefroot Walter	1008	0.153
Van Springel Herman	942	0.150
Pollentier Michel	852	0.117
De Vlaeminck Erik	718	0.070
Demeyer Marc	638	0.151
Vermeire Robert	603	0.079
Demol Dirk	522	0.074
Leman Eric	476	0.090
Verbeeck Frans	474	0.088
Dierickx André	442	0.131
Planckaert Eddy	441	0.063

this approach and its reductionist nature, however, certain physical characteristics are often strongly associated with the male or female gender, making binary categorization still a useful metric.

Because some years contained few pictures, we grouped the images into ten-year periods, starting from 1980 until 2019, the last year of the collection. The first five years of the collection were excluded (1975-1979) so that every period spans a decade. For each period, the ratio of male and female predicted faces was calculated. Figure 2.16 shows this distribution throughout the years. We see that the ratio of predicted female faces steadily rises throughout the years, from around 9% to 27%. These are the predicted genders, so some mistakes will be included in this aggregate, but it still demonstrates the overall trend of the collection.

2.3.10 Comparison with FaceNet

Throughout the FAME project, face recognition models from InsightFace were used. After gathering the prediction labels, a comparative study was performed between the embeddings generated from InsightFace and FaceNet, another pop-

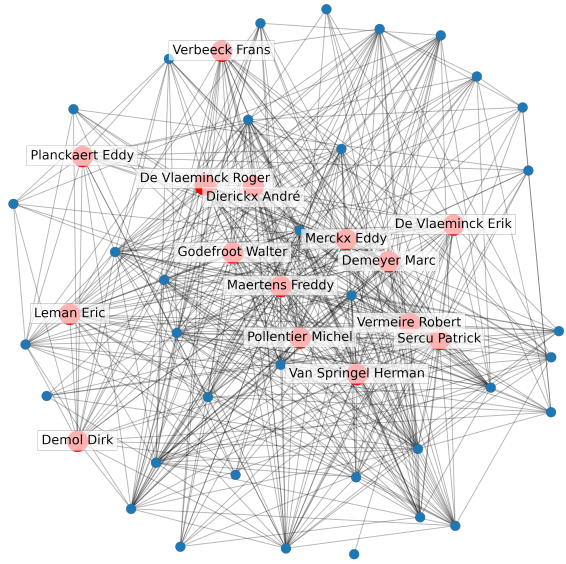


Figure 2.15: Subgraph of the top-15 most frequently identified persons (in red) and their connections.

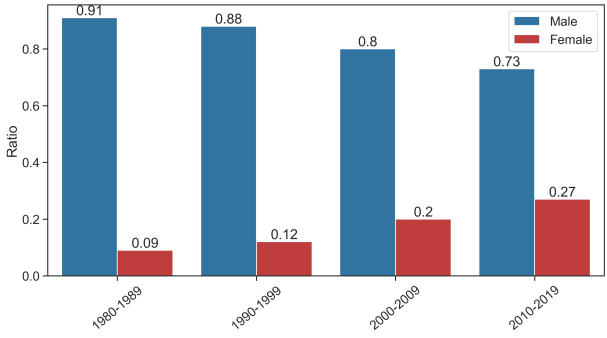


Figure 2.16: Ratio of male and female predicted faces, grouped per decade, for the collection of the Flemish Parliament.

ular open-source face recognition library. For FaceNet, we used an InceptionResnetV1 model that was trained on VGGFace2 [5]. A sample of faces from the dataset was taken, containing the reference images and accepted validated predictions for the Government collection. This set included a total of 17,998 faces. All persons who appeared less than three times were discarded, leaving 17,901 faces.

[5] https://github.com/timesler/facenet-pytorch

For each detected face, the FaceNet embedding was extracted. Next, a leave-one-out test was performed using the InsightFace and FaceNet embeddings. For each embedding, we found the most similar match with all other embeddings via the cosine similarity. If this match belonged to the same person, it was marked as correct and the similarity score was saved. We also saved the similarity score for the most similar incorrect match. We used the Annoy library to quickly find the most similar matches. Figure 2.17 shows the similarity score distributions for both InsightFace and FaceNet embeddings, for the best match and the best incorrect match. The overlap between both distributions visualizes the number of errors made (the best match was a different person). The recognition accuracy for InsightFace was 0.989 (188 errors) and 0.959 (736 errors) for FaceNet. Clearly, the newer InsightFace model outperforms FaceNet, which was expected. We also see that the average correct prediction score is much higher for FaceNet, which means a larger threshold should be used when making person predictions with FaceNet compared to InsightFace.

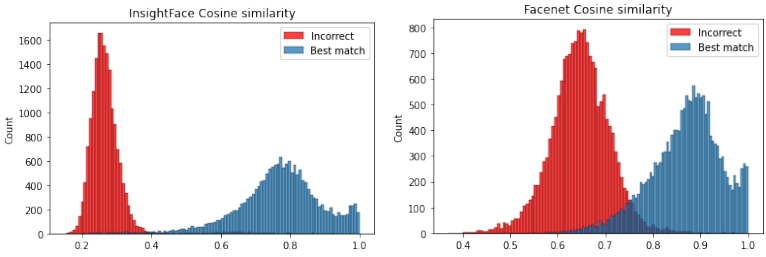


Figure 2.17: Similarity score distributions of the best match (highest similarity score) and best incorrect match for both InsightFace (left) and FaceNet (right) embeddings. The overlap of the distributions denotes the number of errors made.

2.3.11 Discussion

This section presented a facial recognition pipeline that was used successfully in the FAME project. The pipeline was able to accurately recognize persons in each image using a reference set of known persons. The construction of the reference set required the most manual effort and was crucial to make good predictions. It should consist of persons of interest likely to appear in the archive collections. For each person, a handful of images (3-5) was sufficient for robust recognition. It is important that these images are of a single person, of good enough quality, and that the person is facing the camera. We have noticed that low-resolution, noisy images, tend to match with a higher score with other noisy images. Furthermore,

relatively small or extremely large faces on such noisy images were often not detected, which was also noted in [33].

It's also recommended to include images in the reference set from a similar time period as the collection. Reference images with a similar resolution or noise (e.g. from digitization) as those in the collection will generally result in a higher matching score. Furthermore, as people grow older, their facial features change, which affects the embedding vector. For several persons in the reference set, only more recent reference images were used. Surprisingly, some of those persons are still accurately predicted on archive images taken decades ago. An example is shown in Figure 2.18. Even though all the reference set images for this person were taken two decades later, the prediction is relatively accurate with a score of 0.69. If we then look at the highest score between this image and a sample of other archive images for this person from the same period, we see the similarity scores are slightly higher at 0.77, demonstrating the importance of a more varied reference set.

Figure 2.18: Left: Sample image from the reference set for Bert Anciaux. Right: Sample image from the ADVN collection depicting a young Bert Anciaux. Even though all the reference set images for this person were taken over 20 years later, the prediction is relatively accurate with a score of 0.69.

After extracting the embeddings from the reference set and archive dataset, these were matched using cosine similarity. In this work, we mainly focused on the best-matching person prediction. But remember that for each face we have calculated the similarity with all faces in the reference set. Persons in the reference set also have multiple images and embeddings. Therefore, the person prediction is a list of names containing duplicates, ordered by similarity. We are

confident the results can be further improved using the top-k person predictions. If these predictions are all the same person, they could be viewed as confident, even with a lower similarity score threshold.

As metadata enrichment is our end goal, precision should likely be prioritized over recall. Only confident predictions should be imported back into the archives without human validation. It is arguably better to miss out on some predictions than to incorrectly predict the visible persons on a given image. However, the threshold should not be too high, otherwise, most of the predictions will be discarded. The similarity score itself could also be used to query for persons in the archives, allowing the user to decide whether or not to include non-confident predictions. However, even with a relatively low threshold of 0.5 and no fine-tuning of the pre-trained models, over 93% of the person predictions are correct. This makes facial recognition a good choice to enhance the accessibility of archive collections, with minimal manual input. It only requires a handful of images per person. As many different institutions have images featuring the same persons, it could prove useful to collaborate and exchange reference sets or embeddings. Such collaborations would greatly reduce the amount of manual effort required to implement a facial recognition system.

Frequently occurring persons that were not included in the reference set were also detected via DBSCAN clustering of their embeddings. The results of this clustering depend on the parameters, namely the epsilon (controls how dense each cluster needs to be) and the minimum number of faces per cluster. Our initial test using epsilon as 0.5 and 5 faces per cluster produced good results. Usually, most clusters contained images from a single person. Decreasing the epsilon parameter will make the clustering more strict, reducing the total number of clusters and average images per cluster, but it will also reduce the number of errors (multiple persons inside one cluster). Besides clustering, the number of detected faces per image can be used to separate group pictures easily. The co-occurrence of multiple persons can also be a useful feature in finding relations between persons. Furthermore, while sometimes less accurate, the predicted age and gender can also be used to refine searches.

The same facial recognition pipeline can also be applied to video. The naive solution to process every frame will produce optimal results, but will also be the most time-consuming. Our initial test using a scene detection model and extracting a set number of frames worked well, but it can still be improved. Using quality metrics to detect the blurriness of each frame, can help to replace bad frames with those of a higher quality. The prediction confidence could also be improved, by looking at succeeding frames and their predictions. If the same person is predicted multiple frames in a row, this prediction is likely more confident.

While facial recognition is a valuable tool for metadata creation, it involves some legal and ethical concerns. First, the processing of biometric data with a

view to identification is strictly regulated under the European GDPR laws. It is important to maintain the right balance for the privacy and rights of those involved, and that the necessary technical and organizational safeguards are in place. Second, it's important to ensure that we apply artificial intelligence in an ethically responsible way in the cultural heritage sector. In this research, we only used public figures in the creation of our reference sets, which reduced the number of legal and ethical barriers, and limited any impact on private individuals. Furthermore, racial and gender biases can be particularly prevalent within facial recognition applications, and there's a risk they could reinforce or increase existing social inequalities. While it isn't completely possible to remove bias from the used technologies, we need to limit any consequences and provide visibility for the potential biases as much as possible. For additional details regarding the legal and ethical concerns and our approach to minimize the negative impacts, see our tech blog [6].

2.4 Conclusion

This chapter discussed how pre-trained object detection and segmentation models can be used with little adjustments to detect common objects. Through a case study, we showed how image retrieval and multimodal similarity models can be used to query picture collections using similar images and natural language. While such methods require additional processing, they enable novel and intuitive ways to explore collections and enhance them with additional metadata.

Furthermore, we have successfully implemented a facial recognition pipeline to enrich archive image collections using pre-trained open-source models. Persons were identified by matching face embeddings from the reference set with the archive collections using cosine similarity. A total of 182 thousand detected faces were manually labeled with a custom labeling tool to validate the predictions. With a minimum similarity of 0.5, the face recognition model achieved a precision of 0.936. With this threshold, we were able to automatically identify over 62 thousand persons depicted in the image archives. By clustering the face embeddings, we found an additional 95 frequently occurring persons that were not initially in the reference set. In summary, computer vision techniques are an effective way to greatly improve the quality and accessibility of archive collections with minimal manual annotation efforts. The resulting metadata allows for a more comprehensive and efficient analysis of the collection.

[6] https://meemoo.be/en/publications/facial-recognition-what-are-the-legal-and-ethical-aspects

References

[1] V. Mohanty, D. Thames, S. Mehta, and K. Luther. *Photo sleuth: Identifying historical portraits with face recognition and crowdsourced human expertise*. ACM Transactions on Interactive Intelligent Systems (TiiS), 10(4):1–36, 2020.

[2] Y. Zhao and Q. Zhu. *Evaluation on crowdsourcing research: Current status and future direction*. Information Systems Frontiers, 16(3):417–434, 2014.

[3] T. Causer and M. Terras. *Crowdsourcing Bentham: beyond the traditional boundaries of academic history*. International Journal of Humanities and Arts Computing, 8(1):46–64, 2014.

[4] G. Colavizza, T. Blanke, C. Jeurgens, and J. Noordegraaf. *Archives and AI: An Overview of Current Debates and Future Perspectives*. J. Comput. Cult. Herit., 15(1), dec 2021. Available from: https://doi.org/10.1145/3479010, doi:10.1145/3479010.

[5] J. Buolamwini and T. Gebru. *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*. In S. A. Friedler and C. Wilson, editors, Proceedings of the 1st Conference on Fairness, Accountability and Transparency, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 23–24 Feb 2018. Available from: https://proceedings.mlr.press/v81/buolamwini18a.html.

[6] P. Terhörst, J. N. Kolf, M. Huber, F. Kirchbuchner, N. Damer, A. M. Moreno, J. Fierrez, and A. Kuijper. *A Comprehensive Study on Face Recognition Biases Beyond Demographics*. IEEE Transactions on Technology and Society, 3(1):16–30, 2022. doi:10.1109/TTS.2021.3111823.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *Imagenet: A large-scale hierarchical image database*. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[8] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. *Superglue: Learning feature matching with graph neural networks*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4938–4947, 2020.

[9] S. Ren, K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*. Advances in neural information processing systems, 28, 2015.

[10] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. *Panoptic segmentation*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9404–9413, 2019.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. *Microsoft coco: Common objects in context*. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[12] G. Jocher, A. Chaurasia, and J. Qiu. *YOLO by Ultralytics*, January 2023. Available from: https://github.com/ultralytics/ultralytics.

[13] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2, 2019.

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. *Arcface: Additive angular margin loss for deep face recognition*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4690–4699, 2019.

[15] F. Radenović, G. Tolias, and O. Chum. *Fine-tuning CNN image retrieval with no human annotation*. IEEE transactions on pattern analysis and machine intelligence, 41(7):1655–1668, 2018.

[16] S. Verstockt, S. Nop, F. Vandecasteele, T. Baert, N. Van de Weghe, H. Paulussen, E. Rizza, and M. Roeges. *UGESCO-A hybrid platform for geo-temporal enrichment of digital photo collections based on computational and crowdsourced metadata generation*. In Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection: 7th International Conference, EuroMed 2018, Nicosia, Cyprus, October 29–November 3, 2018, Proceedings, Part I 7, pages 113–124. Springer, 2018.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781, 2013.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. In M. Meila and T. Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. Available from: https://proceedings.mlr.press/v139/radford21a.html.

[19] J. Li, D. Li, S. Savarese, and S. Hoi. *Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models*. arXiv preprint arXiv:2301.12597, 2023.

[20] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, pages I–I. Ieee, 2001.

[21] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. *Retinaface: Single-shot multi-level face localisation in the wild*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5203–5212, 2020.

[22] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann. *MediaPipe: A Framework for Building Perception Pipelines*, 2019. arXiv:1906.08172.

[23] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. *Joint face detection and alignment using multitask cascaded convolutional networks*. IEEE signal processing letters, 23(10):1499–1503, 2016.

[24] M. Wang and W. Deng. *Deep face recognition: A survey*. Neurocomputing, 429:215–244, 2021. Available from: https://www.sciencedirect.com/science/article/pii/S0925231220316945, doi:https://doi.org/10.1016/j.neucom.2020.10.081.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. *Deepface: Closing the gap to human-level performance in face verification*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1701–1708, 2014.

[26] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. *Labeled faces in the wild: A database forstudying face recognition in unconstrained environments*. In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.

[27] F. Schroff, D. Kalenichenko, and J. Philbin. *Facenet: A unified embedding for face recognition and clustering*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[28] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. *Vggface2: A dataset for recognising faces across pose and age*. In 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), pages 67–74. IEEE, 2018.

[29] D. Yi, Z. Lei, S. Liao, and S. Z. Li. *Learning face representation from scratch*. arXiv preprint arXiv:1411.7923, 2014.

[30] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du, and J. Zhou. *WebFace260M: A Benchmark Unveiling the Power of Million-Scale Deep Face Recognition*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10492–10502, June 2021.

[31] *InsightFace: State of the art deep face analysis library*, 2022. Available from: https://github.com/deepinsight/insightface.

[32] M. R. Kale, P. P. Rege, and R. D. Joshi. *Designing a Dual-Level Facial Expression Evaluation System for Performers Using Geometric Features and Petri Nets*. J. Comput. Cult. Herit., feb 2023. Just Accepted. Available from: https://doi.org/10.1145/3583557, doi:10.1145/3583557.

[33] M. Wevers and T. Smits. *Detecting Faces, Visual Medium Types, and Gender in Historical Advertisements, 1950–1995*. In A. Bartoli and A. Fusiello, editors, Computer Vision – ECCV 2020 Workshops, pages 77–91, Cham, 2020. Springer International Publishing.

[34] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. *DSFD: dual shot face detector*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5060–5069, 2019.

[35] P. Lisena, J. Laaksonen, and R. Troncy. *FaceRec: an interactive framework for face recognition in video archives*. In DataTV 2021, 2nd International Workshop on Data-driven Personalisation of Television, 2021.

[36] J. Guo, J. Deng, A. Lattas, and S. Zafeiriou. *Sample and Computation Redistribution for Efficient Face Detection*. In International Conference on Learning Representations, 2022. Available from: https://openreview.net/forum?id=RhB1AdoFfGE.

[37] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, volume 96, pages 226–331, 1996.

[38] B. Castellano. *PySceneDetect, Intelligent scene cut detection and video splitting tool.*, 2022. Available from: https://scenedetect.com/en/latest/.

[39] K. Milleville, A. Van den Broeck, R. Vissers, B. Magnus, N. Vanderperren, A. Vergauwe, E. Van Keer, T. Ruette, and S. Verstockt. *FAME video browser – face recognition based metadata generation for performing art videos*. In DH Benelux 2022-ReMIX: Creation and alteration in DH (hybrid), pages 1–4. Zenodo, May 2022. Available from: https://doi.org/10.5281/zenodo.6517242, doi:10.5281/zenodo.6517242.

[40] S. Verstockt, K. Milleville, D. Ali, F. Porras-Bernandez, G. Gartner, and N. Van de Weghe. *EURECA: EUropean Region Enrichment in City Archives and collections*. In 14th ICA conference: Digital approaches to cartographic heritage, pages 161–169. Aristoteleio Panepistimio Thessalonikis (APTh), 2019.

3

Automatic Processing of Raster Maps

"Theory will take you only so far"

– J. Robert Oppenheimer

This chapter discusses automated processing methods for digitized raster maps. First, the challenges and related work are discussed. Next, a custom method for automated geolocalization is presented. The chapter finishes by presenting road segmentation approaches on topographic and walking maps.

This chapter features an adapted version of the following publication:

Milleville, K., Verstockt, S., & Van de Weghe, N. (2022). **Automatic Georeferencing of Topographic Raster Maps.** ISPRS INTERNATIONAL JOURNAL OF GEO-INFORMATION, 11(7). https://doi.org/10.3390/ijgi11070387

3.1 Introduction

The digitization of historical maps has given researchers access to high-quality geographical data from the past. These maps are often the only digitized source of reliable data, making them a valuable resource. Many institutions have digitized large collections of raster maps depicting different time periods and locations.

This time-consuming digitization process is usually performed manually. Most maps contain little metadata, such as a generic title, date, and a short description. This makes it difficult to efficiently search for maps that describe a specific region of interest. Querying on the name of that region will not return complete results, as most place names do not appear in the title or description of the map. Creating additional metadata for these collections will greatly improve their accessibility and provide new opportunities for novel research.

Manually annotating or georeferencing these collections can be a tedious process. Therefore, institutions mainly focus on the most important items in their collections. Crowdsourcing approaches are often used to annotate these collections with valuable metadata. For raster maps, an interactive program is typically used to manually georeference the maps [1–3]. Users need to select matching control points on both the raster map and at the corresponding locations on Earth. The program then automatically georeferences and corrects the raster map. This technique provides accurate results, given that the control points are selected correctly. Once the maps in the collection have been georeferenced, they can be queried for specific regions of interest. However, searching for toponyms is still not efficient, as each map must be manually checked for the desired toponyms.

To query the toponyms present on the map, these have to be annotated. Manual annotation can take up to several hours for one map, which is not feasible for large collections. Therefore, text detection and recognition approaches can be used to automatically detect and transcribe the text present on the maps. Compared to a traditional optical character recognition (OCR) approach for scanned document images, where the text is structured in horizontal lines and paragraphs, raster maps come with additional challenges. Text labels can be handwritten, appear in different orientations, sizes, fonts, and colors, overlap one another, and even curve along with the described geographical features (e.g., rivers) [4]. Additionally, historical maps can be degraded or digitized at a lower resolution, further reducing the transcription accuracy [5]. Figure 3.1 shows part of a historical United States Geological Survey (USGS) topographic map from 1886[12]. It features different text styles and complex text placements, which are common for historical raster maps.

The recognized text can be linked to the correct contemporary toponyms via publicly available geocoders. These geocoders contain millions of toponyms and attempt to match an input string with their database. To account for spelling errors, fuzzy matching is often used. Fuzzy matching includes non-exact matches and is necessary for historical maps because some toponyms are spelled differ-

[1] https://www.usgs.gov/programs/national-geospatial-program/historical-topographic-maps-preserving-past

[2] https://github.com/spatial-computing/map-ocr-ground-truth/tree/master/USGS-60-CA-modoclavabed-e1886-s1884

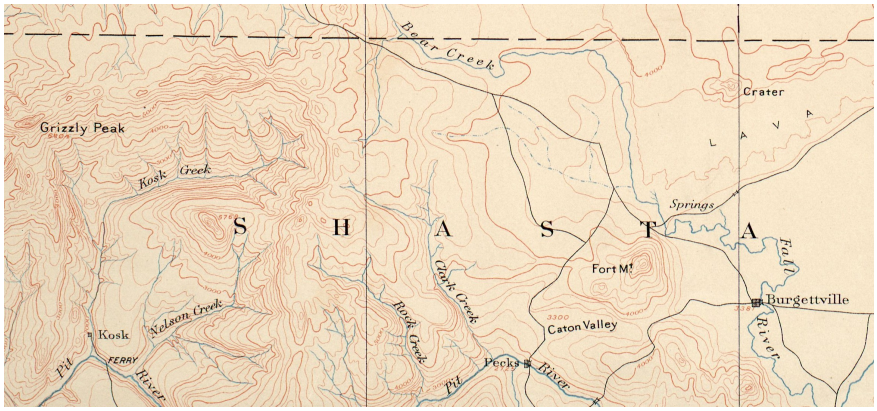


Figure 3.1: Crop of a USGS topographic from 1886 featuring parts of California. The visible texts have different fonts, sizes, and curve along geographical features.

ently over time. Linking the recognized text to geocoders and open data improves map accessibility and reduces recognition errors by eliminating false positives.

Besides toponyms, raster maps typically contain a representation of existing geographical features, such as roads, waterways, vegetation, etc. Contemporary raster maps are generated from existing geolocated vector datasets containing these features and visualized using a specific set of rules [6]. However, historical raster maps generally do not have associated vector data. This makes it very difficult to perform research on land use or other geographic features using these historical maps. Computer vision methods can aid in processing these maps to automatically segment and vectorize the visible geographical features.

3.2 Related Work

Text recognition is a major field of computer vision and has been the focus of many research papers and studies. With the rise of convolutional neural networks, supervised machine learning techniques became state of the art for OCR and (scene) text recognition. OCR results on scanned document images from state-of-the-art and commercial tools are generally excellent and depend mostly on the quality of the scan [7]. Text recognition on natural images is generally harder, as these usually contain a larger variety of text fonts and backgrounds. With the rise of larger and more varied datasets (e.g. ICDAR2015 and Coco-Text [8, 9]), state-of-the-art text detection and recognition models can already achieve a relatively high accuracy [10]. Many of those works publish their pretrained models and the code needed to use them. However, when using these models on raster maps, both the

text detection and recognition performance are generally worse, even when using commercial recognition services [11]. This decrease in performance is mainly due to the higher complexity of backgrounds and text label placements for raster maps, compared to natural images.

Because of this lower accuracy on raster maps, pre- and post-processing techniques are frequently used to improve the results. A common approach is to first extract the text labels from the map and afterward perform text recognition. A combination of computer vision techniques, such as connected components analysis and color quantization can be used to differentiate the text labels from the background of the raster maps. These techniques all generate similar results, namely binarized images that are easier to process [12, 13]. Because it can be difficult to automatically differentiate the foreground text from the background, a semi-automatic approach can be used to improve the results. Chiang et al. [14] developed a general, semi-automatic text recognition technique, where users needed to label a small number of crops on the maps and indicate whether they contained text. They then used computer vision techniques to homogenize multi-oriented and curved text. These preprocessing techniques greatly increased the recognition accuracy of the used OCR software.

After recognizing the text on the map, geocoders can be used to match the text labels to the corresponding toponyms and their coordinate locations. These coordinates can then be used to estimate the correct geolocation of the map. However, many problems occur when matching the recognized text with the correct toponym. The spelling of a toponym may have changed over time and the recognition likely contains errors. Therefore, fuzzy string matching is required to deal with these spelling errors. However, the ambiguity of place names, further increased by fuzzy matching, can produce many false positives. When queried for a given string, a geocoder may return a multitude of toponyms from different countries. This makes it difficult to determine which match, if any, is correct for the given text label. These problems are further amplified when common street names or points of interest (e.g., Main Street, church) are recognized. The toponym may not even be present in the geocoder's database, a possibility that is further increased for older maps. Another possibility is that the toponym is not detected at all. However, since topographic maps generally contain many toponyms that are relatively close to each other, most false positives can be filtered out and a general location of the map can be estimated.

Weinman [15] used this information along with known toponym geocoordinates and feature label placements to construct a probabilistic model that improved text recognition accuracy. He was able to reduce the word error rate by 36%, compared to the raw OCR output. In [16], the framework for an automated open-source map processing approach is presented. Part of the framework includes a module for automatic geolocalization. As a preliminary result of this

module, the first toponym match for a label with a confidence value of 90% or more obtained from the Google Geocoding API was used to geolocate each map. The geocoding results were clustered and the centroid of the largest cluster was used to predict the actual center of the map. The approach was then validated on 500 randomly selected maps from the NYPL georectified map collection[3]. The results were promising: 37% of the geolocated maps were within a 15 km radius from the ground truth coordinates and 28% were within a 5 km radius. The authors noted that considering only a single toponym match for each text label was very restrictive and that the text recognition sometimes failed to detect enough text labels. In a follow-up work by the same authors, they developed a text linking technique that further improved results [17]. By using both the textual and visual content, they were able to train a model that could correctly link multiple words of the same location phrase (e.g., linking "Los" and "Angeles" to "Los Angeles"). The text linking greatly reduced the number of false-positive geocoder matches and made the geolocation more precise.

Our goal was to provide a general geolocation technique based on text recognition results from topographic maps. We used pretrained text detection and recognition models, to show that it is possible to accurately geolocate and annotate topographic raster maps without the need for labeled datasets and custom models. This makes our approach useful to many institutions and researchers who wish to annotate their collections of digitized raster maps with minimal manual input. The text detection and recognition models can be replaced by other text recognition services. The pretrained model[4] uses the same text detector as in [18] and the same recognition model as in [19]. The main benefit of using this detection model as opposed to another popular text detector such as EAST [20], is that this model was trained to detect text on a character level, providing more flexibility for rotated and curved text.

Many approaches exist for the segmentation of roads from raster maps. These range from traditional computer vision methods, to using convolutional neural networks [21]. Chiang et al. [6] presented an end-to-end road segmentation approach from raster maps. Their approach consisted of three major steps. First, they extracted the road geometry. Next, the road intersections were detected. Finally, the result was vectorized. They used a variety of computer vision techniques, which included most notably color quantization. While their approach produced great results, it still required some user input, to select the road colors in the given image. In [22], the authors proposed a vector-to-raster alignment algorithm to annotate geographic features on raster maps. This way, existing spatial data can be used to train CNNs in a weakly-supervised way. Their algorithm is generic,

[3] http://spacetime.nypl.org

[4] https://github.com/faustomorales/keras-ocr

and it can be applied to multiple geographic features. In [23], the authors proposed a domain adaptation technique and used it to train semantic segmentation models for hydrological features on historical maps. They introduced a novel loss function that corrects for object changes and spatial misalignment. Their experiments showed that their approach outperformed the state-of-the-art, even with limited supervision. For a more detailed overview of different road segmentation techniques, see [21].

3.3 Automated Geolocalization

This section proposes a generic pipeline to georeference topographic raster maps and extract the visible text and toponyms as linked open data (LOD). An overview of the entire pipeline is given in Figure 3.2. The first step in the pipeline is to preprocess the raster maps and extract the actual map region. Most digitized raster maps contain additional information outside the map boundary that can affect geolocation accuracy. The second step of the pipeline details text recognition and geocoding. Next, both the location of the recognized text (pixel coordinates) and the location of the matching geocoding results (geocoordinates) are used to estimate a geolocation for the map. After recognizing and geocoding the text on each map, each text label was matched with a list of possible geolocation coordinates. To georeference the maps, we need to find control points on the map itself and their corresponding WGS84 coordinates (latitude and longitude). Using these point pairs (matching pixel and geocoordinate pairs), a transformation can be calculated to convert pixel coordinates to geocoordinates and vice versa. We have developed an iterative algorithm that uses the text label locations and their geocoder matches, to generate four control points representing the four map corners in WGS84 coordinates. These are then compared to the correct corner geocoordinates, to estimate the accuracy of our method. The map geolocation was determined in multiple steps, by first estimating an initial region of interest (ROI). This ROI was then further refined by iteratively removing outliers via a RANSAC filtering approach. In each step, we filtered out the geocoder matches that had a low probability of being correct. The new ROI was chosen as the bounding box of the geocoordinate matches that remained after filtering. This ROI was subsequently refined until no more outliers were found. Finally, the locations of the text labels on the raster map and their matching geocoordinates were used to calculate the four control points and georeference the map. The output can be saved as GeoJson or any other commonly used format and contains the estimated map geolocation, the recognized text on the map, and found toponyms as LOD.

The approach does not depend on how the text labels and geocoder matches were generated, therefore it can be used with any text recognition system and

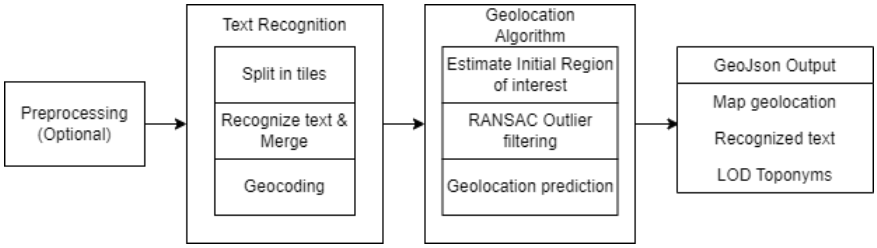


Figure 3.2: Overview of the geolocation pipeline

geocoder. The geolocation accuracy largely depends on the text recognition accuracy and geocoder results. If the map is of low quality and most text labels are not recognized, the results will be poor. For each text label, we define its location on the map as the center point of the text detection bounding box. All figures presented in this section refer to the same topographic map of Gent-Melle, from dataset M834 (see 3.3.9.1). The code is made publicly available at https://github.com/kymillev/geolocation.

3.3.1 Preprocessing

The dataset of historical topographic map sheets from Belgium contains additional information outside the map boundary. Each map is surrounded by a black border and some blank space, in which coordinate information is given. At each corner, the map is georeferenced based on the 1972 Belgian Datum. The numbers surrounding the map denote the X and Y coordinates in the Belgian Lambert72 projection [24]. This dataset was first preprocessed and the effective map region was determined. It is not strictly necessary to extract the effective map region to geolocate the map. It does make the used techniques slightly more accurate, as the image crop now only contains the map itself, but not the legend, surrounding coordinates, and toponyms. These labels could be incorrectly recognized as toponyms, leading to additional false positives. Figure 3.3 shows the upper-left corner of one of the topographic maps and the extracted map region.

Morphological operations (erosion and dilation) were used to detect the thick black borders surrounding each map. Next, small crops were taken along the edges of the map. The text within these crops was recognized. For both the left and right side edges, the crops were first rotated so that the X/Y coordinates were upright. We used the location of these surrounding text labels to determine the effective map region. Detection of this inner region with morphological operations alone was inconsistent due to the thin outer edges and slight rotations of some of the scanned maps. After recognizing each text label, the Lambert coordinates

were filtered out and the remaining text was used to extract the effective map region. The dataset of contemporary topographic maps did not contain any additional information surrounding the maps and was therefore not preprocessed.

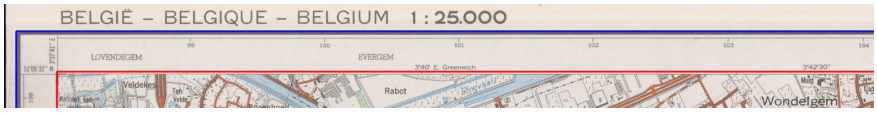


Figure 3.3: Outer rectangle detected by morphological operations (in blue), and the effective map region determined via text recognition on the surrounding coordinates (in red).

3.3.2 Text Recognition

Since the images of the map sheets have a much higher resolution than typical images, the performance of the text detection model on the entire dataset was poor. Smaller text labels were consistently not detected when using the full-sized maps. This is likely due to the text detection model using global thresholds to segment the text regions. Therefore, a tiling approach was used to improve results. Previous work showed that larger tile sizes are preferred over smaller ones [11]. Each image was divided into multiple tiles of 2500x2500 pixels, with an overlap region (in both X and Y) of 500 pixels. Text detection and recognition were performed on each of the tiles. Text labels detected in the overlap region were merged with overlapping and similar text labels from adjacent tiles to avoid splitting words at the edges of each tile. After merging, the recognized text was post-processed.

First, the text labels that only contained digits were filtered out. These labels denoted height contours, kilometer milestones, or highway segments and did not provide meaningful results in the following geocoder steps. Next, overlapping detections of multiple text labels were merged, as many toponyms consist of multiple words. This can introduce additional errors, by merging incorrectly. Therefore, overlapping detections were only merged if their relative orientations were within 15 degrees of each other. We found that this threshold eliminated most of the incorrectly merged labels. Minor problems still occasionally occurred due to text detection errors and complex arrangements of toponyms. Figure 3.4 shows one of these complex arrangements, where each detected label overlaps with another and also shows the result after merging.

When merging multiple overlapping text labels, we sorted them in the natural reading direction. The individual detections were first sorted from top to bottom and divided into different groups based on the difference in their Y-coordinates.

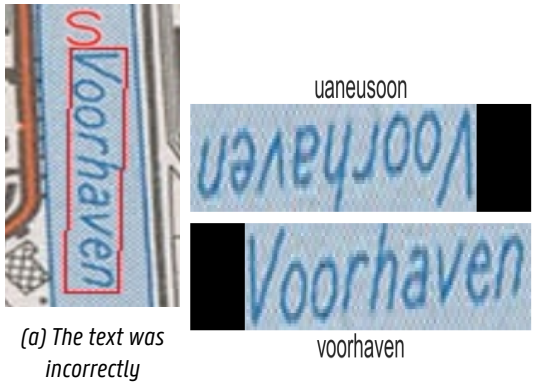
Then, each group was sorted individually from left to right, resulting in the natural reading direction. Only text labels (usually denoting rivers) that were read from bottom-left to top-right, were sometimes merged incorrectly.

Figure 3.4: Example of merging the text labels in the natural reading direction.

Vertically oriented text was usually detected correctly but often transcribed incorrectly, as can be seen in Figure 3.5a. Because the recognition model assumes that the text is oriented from left to right, such errors occur. After detecting the effective text region, the image crops were warped into horizontally oriented text. If the label was vertically oriented, this transformation needed to be adjusted to warp it correctly. If the leftmost point of a text label is on top, the text is normally read from top-to-bottom. Due to minor errors in text detection, we cannot always rely on the coordinates of the predicted bounding boxes, so two warping transformations are possible. Both image transformations were performed and the text was recognized. In a later step, the incorrect prediction was filtered out using the geocoder results. Figure 3.5b shows an example of the proposed solution. We suppose a more elegant solution can be used to determine the correct text orientation, based on the visual information or the text content. Because each map only contains a handful of such vertical text labels, such an improvement was left as future work. As the main goal of this work is to show how off-the-shelf text detection and recognition models can be used to effectively georeference topographic raster maps with little adjustments, no additional processing or linking of the detected text labels was performed.

3.3.3 Geocoding

After recognizing and processing the text labels on the map, multiple geocoders were queried with each predicted text label. Strings shorter than three characters were ignored as they rarely returned meaningful results. Three different geocoding services were used: Google Geocoding, TomTom Geocoding, and Geonames (open source). We originally intended to only use Geonames but found that many



(a) The text was incorrectly recognized as the character "s".

(b) Text recognition result after warping the text in both ways.

Figure 3.5: An example of the proposed solution for vertically oriented text.

queries did not return meaningful results, while good matches were found with the commercial geocoders. For each geocoder, the resulting toponyms were compared to the query string via the partial string similarity score[5]. Given two strings of length n and m, if the shorter string is length m, the partial string similarity will return the similarity score (based on the Levenshtein distance[6], a popular string similarity metric) of the best matching length-m substring. This similarity score ranges from 0 (mismatch) to 100 (perfect substring match). We found that this score performed better than the standard Levenshtein distance, especially for shorter strings. For each match, the toponym name, geocoordinates, similarity, and type (populated place, street, etc.) were saved. Because we used multiple geocoders, these results contained duplicates. The duplicates were removed if there was an exact match for the toponym name and type and if both coordinates were close to each other. Checking their closeness is important, as two differently located places or streets can have the same name. Some duplicate matches still remained, but these had little effect on the final region of interest.

Each map contained an average of 365 and 671 usable text labels, for the Belgian (M834) and Dutch (TOP50raster) datasets, respectively. A histogram of the geocoder matches per text label for both datasets is shown in Figure 3.6. It is clear that the distributions are asymmetric and that most text labels have a small number of matches. The labels with a larger number of matches are the least informative to predict an area of interest, as these denote common place names, spread over a large area.

[5] https://github.com/seatgeek/thefuzz#partial-ratio

[6] https://en.wikipedia.org/wiki/Levenshtein_distance

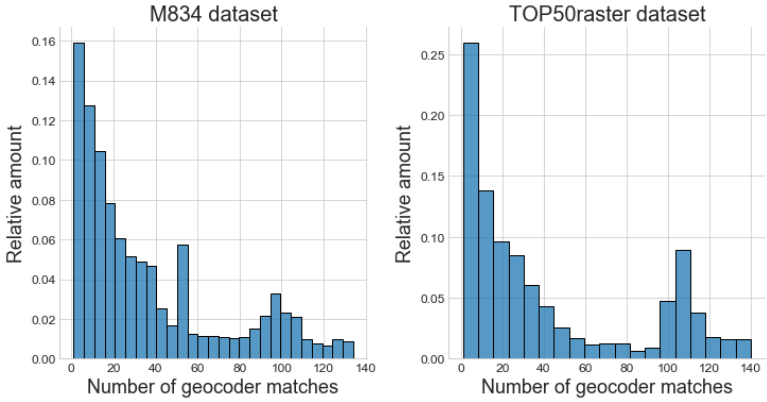
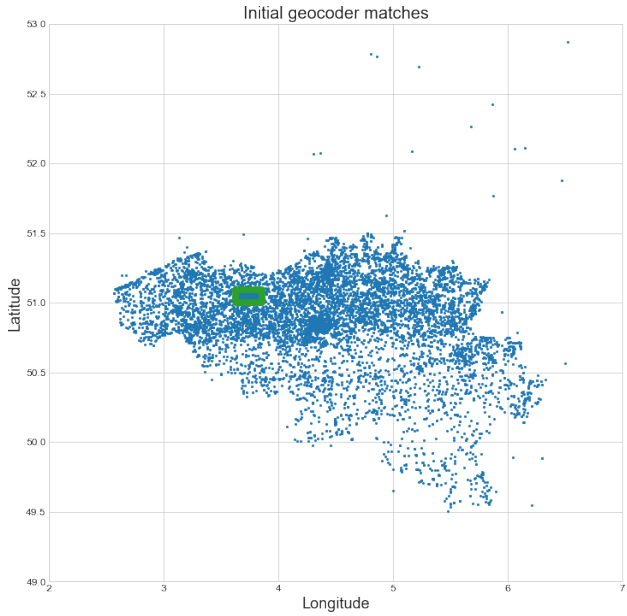


Figure 3.6: Histograms showing the distribution of the number of geocoder matches per recognized text label for both datasets. Many text labels have a large number of geocoder matches (>50), which do not provide much value.

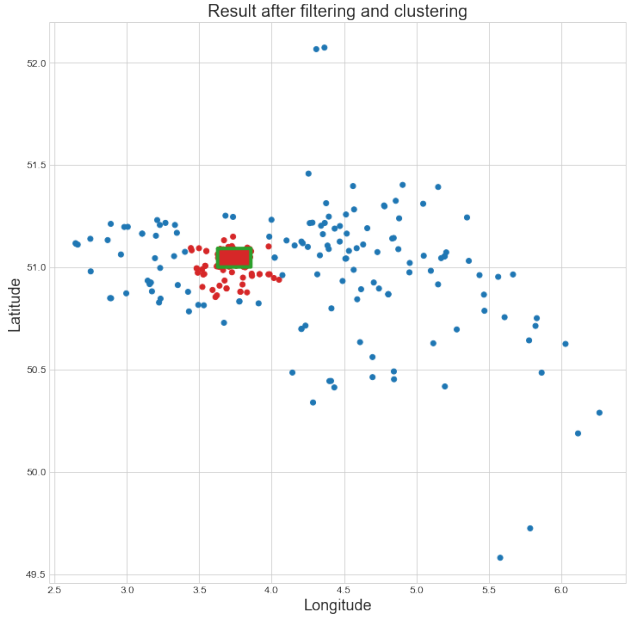
3.3.4 Estimating an Initial Region of Interest

Because some place names are common, certain queries yielded more than 100 geocoder matches. Many street names, such as "Kerkstraat" (comparable to "Main Street" in the USA), are common in Belgian cities. Plotting these coordinates revealed possible matches throughout Belgium and the Netherlands and a small number of random locations around the world. Even though the geocoders allow a country to be specified, the results are not always limited to that country. Most of these geocoder matches were not correct for the queried text. They were either common place and street names or were found due to the fuzzy matching of the geocoders. Figure 3.7a displays the coordinates of all initial geocoder matches for all recognized text labels on the map of Gent-Melle. Clearly, these are distributed all over Belgium, with some outliers.

Text labels with a large number of matches are therefore not very relevant for predicting an initial region of interest. Similarly, geocoder matches with a lower string similarity with the corresponding text label have a lower probability of being correct. Therefore, we discarded geocoder matches with a partial string similarity below 90. Afterward, text labels that contained more than five geocoder matches were also discarded. In this way, the worst string matches and the most common place names were filtered out. These toponyms were still scattered over hundreds of kilometers and clustering them produced huge regions of interest. Therefore, geocoder matches were also filtered by their relative coordinate distance. Assuming that the correct matches were found, their relative distances should be small and they should be distributed relatively uniformly on the map.



(a) Coordinates of the initial geocoder matches. The green rectangle denotes the ground truth geolocation of the map. The shape of Belgium is visible in the distribution of points (extreme outliers are not shown to improve the visibility of the figure).



(b) Result after filtering and clustering of the initial coordinates. The largest cluster found is shown in red. The green rectangle denotes the ground truth geolocation of the map.

Figure 3.7: Coordinates of the initial geocoder matches before and after filtering and clustering.

Subsequently, each correct coordinate should be relatively close to other correct coordinates. Therefore, the haversine distances between the toponym candidates were calculated and a geocoder match was removed if the distance to any of the five nearest neighbors was greater than 100 km or if the distance to the nearest neighbor was greater than 25 km. This additional filtering ensured that clear outliers were removed. These distance thresholds depend somewhat on the scale of the map. However, most topographic maps contain many toponyms, so the relative distances of correct matches should still be small. For the datasets used, a much smaller distance threshold could be chosen, as each map's diagonal only covers approximately 19 and 32 km.

Next, the remaining geocoordinates were clustered with the DBSCAN [25] clustering algorithm, which clusters higher density regions. This clustering eliminated additional outliers and was also used successfully in previous work [16, 17]. We used the reciprocal of the number of geocoder matches as a sample weight for each point. In this way, points with multiple matches received a lower weight in clustering. The bounding box of the found cluster was taken as the initial region of interest. If multiple clusters were found, the one containing the most points was selected. Figure 3.7b displays the result after discarding the low-probability coordinates and clustering. A more detailed plot of the remaining coordinates after determining an initial region of interest for the map of Gent-Melle is shown in Figure 3.8.

3.3.5 Refining the Region of Interest

To refine the initial region of interest, we used both the locations of the text labels and the coordinate locations of their geocoder matches. Generally, the relative pixel location on the map should be very similar to the relative coordinate location of the corresponding toponym match. Similarly, text labels that are further away from each other on the map should have corresponding geocoordinates that are also further away from each other. Naturally, the exact location of the text label will not fully correspond with the correct geocoder coordinates. However, outliers can be filtered out, as this error is quite large on average for incorrect geocoder matches.

3.3.6 Predicting the Geolocation

We used a RANSAC-based approach to remove most false-positive coordinate matches. RANSAC is a generic iterative algorithm that can fit a model while still being robust to outliers [26]. Figure 3.9 gives an overview of the RANSAC-based outlier filtering.

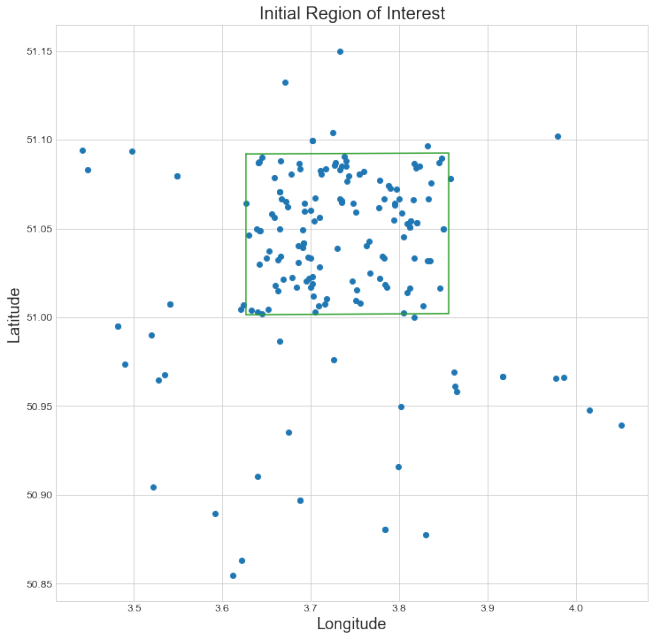


Figure 3.8: Coordinates of the geocoder matches that were located inside the initial region of interest. The ground truth geolocation of the map is shown in green.

First, a set of inlier point pairs was randomly selected from all possible point pairs. Each recognized text label and corresponding pixel location on the raster map was given a 50% probability of being selected. For each selected pixel location, we then randomly chose one of the corresponding geocoordinate matches. This produced a randomized set of point pair inliers. Assuming these inliers were correct, the map geolocation was predicted (see 3.3.6). This geolocation was then used to filter the selection of additional point pairs in the RANSAC algorithm. For each other point pair not initially selected, we performed two checks to decide if these should be added to the set of inliers. First, we checked if the geocoordinates were inside the predicted geolocation. Next, the relative position error (see 3.3.7) was calculated. If the point was inside the geolocation and the error was smaller than a predetermined threshold, we added the point as a candidate inlier. Finally, both the initial inliers and new candidates were used to make a new prediction of the map geolocation and calculate the average relative distance error over all selected point pairs. This entire process was repeated for 10,000 iterations and the set of point pairs with the lowest average error was chosen for further processing.

As mentioned before, the location of a text label on the raster map does not fully correspond with its corresponding toponym geolocation. There is a slight

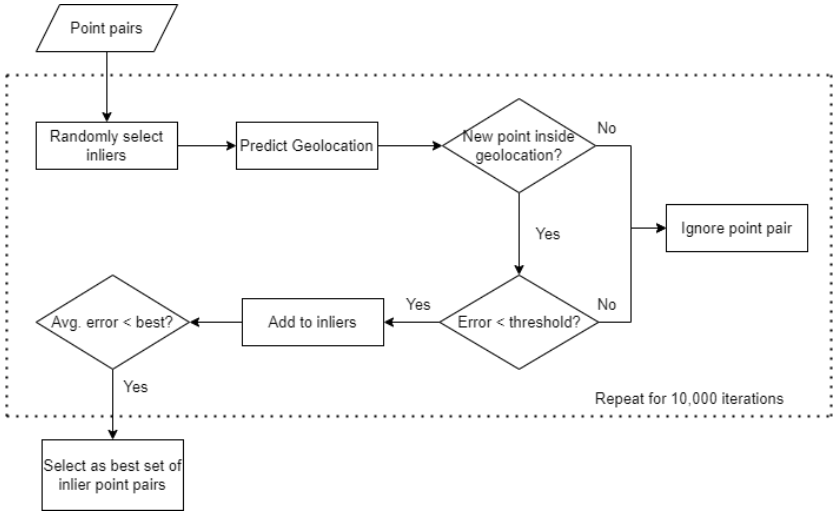


Figure 3.9: Overview of the RANSAC outlier filtering algorithm.

error for each coordinate pair, as the text label placement depends largely on other geographical features or symbols in that location. If the text label overlaps with the other map features, it is typically moved to provide a better view of the landscape. An underlying feature (river, road, etc.) will not be altered so that a toponym label can be placed more correctly. Combined with the fact that we still cannot guarantee which point pairs are correct, we use the average coordinate vector of all selected point pairs to geolocate the map, to reduce the error in label placement. Even if the toponym labels have a consistent bias contrary to our assumptions, for instance, by making the geolocation correspond to the top-left corner of the toponym label instead of its center, this would only result in a minor translation error in our final prediction of the control points.

After randomly selecting an initial set of coordinate pair inliers, we estimate the map geolocation assuming that these pairs are correct. This geolocation is then used to filter the selection of additional coordinate pairs in the RANSAC algorithm. First, the average pixel and geocoordinate vectors were calculated, resulting in a central point pair. Next, the absolute vector differences with each point pair and the centers were calculated. The differences in X and Y were calculated independently, as were the differences in longitude and latitude. Afterward, a linear conversion factor between pixel coordinates and geocoordinates was calculated. All of the differences in X were divided by the differences in longitude, and similarly, all the differences in Y were divided by the differences in latitude for each point pair. We now have calculated a conversion factor from pixels to geocoordinates for each point pair and the center points. Because the text label

position is not perfectly aligned with the corresponding geocoder coordinates, the median for each of these factors was selected, resulting in an average conversion factor from pixel coordinates to geocoordinates. The median is preferred over the mean as it is more robust to outliers. To calculate the conversion factor, we only need two correct point pairs. By taking the median factor of all point pairs, the variance and geolocation error is reduced since it does not fully rely on the correctness of a single pair (which we cannot know). Because the pixel coordinate origin corresponds with the upper left corner of the raster map, the conversion factor for Y must still be multiplied by -1.

After calculating the central coordinate vectors and the conversion factor, the four map corners can be transformed from pixel coordinates to corresponding geocoordinates. Assuming that the predicted geolocation is a rectangle, only the upper left and lower right points need to be transformed to georeference the map. The calculation of the control points is described as a vector equation in Eq. (3.1). Where, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ denote the upper left and lower right control points, respectively, $\boldsymbol{C_{geo}}$ and $\boldsymbol{C_{xy}}$ the average geo- and pixel coordinate vectors, $\boldsymbol{f_{xy}}$ denotes the conversion factor, and $\boldsymbol{B_{xy}}$ is a 1D vector containing the width and height of the raster map.

$$\begin{aligned} \boldsymbol{C_1} &= \boldsymbol{C_{geo}} - \boldsymbol{f_{xy}}\boldsymbol{C_{xy}} \\ \boldsymbol{C_2} &= \boldsymbol{C_{geo}} - \boldsymbol{f_{xy}}(\boldsymbol{C_{xy}} - \boldsymbol{B_{xy}}) \end{aligned} \tag{3.1}$$

Basically, the translation vector of the average pixel coordinate and map boundaries ([0,0] and [w,h]) is calculated and multiplied by the conversion factor to get the corresponding translation vector in geocoordinates. This vector is then subtracted from the average geocoordinate to attain the predicted map boundaries in WGS84 coordinates.

3.3.7 Relative Position Error

In each RANSAC iteration, a set of inlier coordinate pairs was randomly selected and the map region was predicted. For each pair that was not selected, we checked if the geocoordinates were within the predicted map region. If they were, the relative position error was calculated. If this error was smaller than a predetermined threshold, the coordinate pair was added to the set of new candidate inliers. This simple but effective relative position error was calculated by comparing the relative position of each point pair with the inlier point pairs, for both the pixel and geocoordinate locations. We check how many points are to the left/right of the current point on the map and how many points are to the left/right of the associated geocoordinates. We take the difference between these two values and calculate a similar difference for how many points lie above/below the specified

pair. Finally, these differences are added and divided by the number of inlier point pairs. This way, the metric is normalized to the total number of inlier pairs considered. The error metric can be calculated very efficiently and will discard many outliers during the randomized inlier candidate selection. We found good results with a threshold of 0.05. So each candidate inlier pair's relative position needs to "agree" with at least 95% of the initially selected inlier pairs to be selected. After selecting these new candidates, the error metric is calculated for all of the selected point pairs and averaged. The set of point pairs with the lowest average error after 10,000 iterations was then selected for further processing.

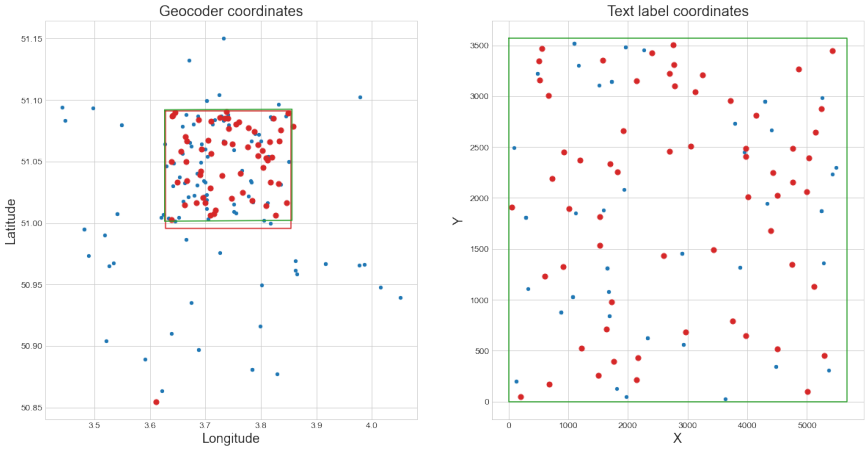


Figure 3.10: Left: Latitude and longitude coordinates of the geocoder matches. The green and red rectangles denote the ground truth geolocation and the predicted geolocation, respectively. Point pairs selected during the RANSAC algorithm are shown in red, the others in blue. Right: X and Y pixel coordinates of corresponding text labels. The green rectangle denotes the raster map bounds (width and height).

3.3.8 Determining the Final Region of Interest

Now that a region of interest has been defined and most of the outlier coordinate pairs have been filtered out with RANSAC, the final geolocation can be estimated. First, the selected coordinate pairs from the previous step were used to predict the map geolocation. This result is shown in Figure 3.10. After geolocating the map, some geocoder matches can still lie outside the predicted region, either because they are incorrect matches or because the predicted region is incorrect. We iteratively deal with such remaining outliers. We find the outlier geocoordinate

point that is farthest from the predicted region, remove that point pair, and predict the geolocation of the map again with all the remaining pairs. We repeat this process until there are no more outliers. The resulting prediction is our best estimate for the map's geolocation. This iterative outlier filtering process further improved the accuracy of our algorithm. Figure 3.11 shows the final geolocation estimate and remaining coordinate pairs for the map of Gent-Melle.

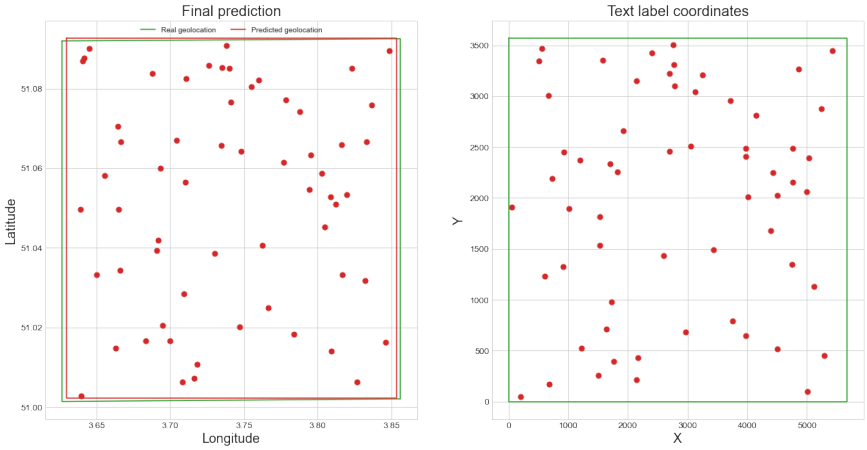


Figure 3.11: Final geolocation prediction (in red) and ground truth geolocation (in green) for the map of Gent-Melle. Point pairs used for the prediction are marked in red. There is a visible correlation between the relative positions of the point pairs.

3.3.9 Evaluation

This section details the two datasets of topographic raster maps used to validate our techniques. We have chosen a dataset of older maps that were later scanned and digitized, as well as a dataset of contemporary maps generated from topographic vector data. We have applied the same techniques to both datasets and have compared our results in section 3.3.9.2.

3.3.9.1 Datasets

M834 topographic raster maps of Belgium
This dataset consists of 16 adjacent topographic map sheets of Belgium, situated around the city of Ghent. These maps are part of the second edition of the

M834 series, and they were created between 1980 and 1987 by the Belgian Nationaal Geografisch Instituut (NGI) [27]. The map sheets have a quality of 225 dpi (6300x4900 pixels), which is lower than the usually recommended quality of 300 dpi for OCR and text recognition. Each map sheet contains a legend and additional information regarding the projection and geodetic system used. The map sheets were printed at 1:25,000 scale in six colors on offset presses. The average length of the diagonal of each map is approximately 19 km. Each map is surrounded by a black rectangle and some blank space, in which coordinate information is given. At each corner, the map is georeferenced based on the Belgian Datum of 1972. The numbers surrounding the map note the X and Y coordinates in the Belgian Lambert72 projection [24]. To validate the georeferencing of these maps, the ground truth WGS84 coordinates of the bounding polygon for each map were taken from the official metadata provided by the NGI. These maps are subject to copyright, so we are unfortunately unable to share the full raster images. However, the maps can be viewed online in Cartesius[7]. A list of all the selected maps is included with our code. Because our georeferencing technique uses the position of the text labels on the map, this dataset was first preprocessed and the effective map region was determined.

TOP50raster

This dataset consists of 9 adjacent contemporary topographic raster map sheets from the full Top50Raster dataset covering the Netherlands. These 9 maps were created in 2018 and are published on PDOK[8], an open-source geospatial data platform, published by the Dutch government. The TOP50raster maps and other raster collections were generalized from the TOP10NL vector data [28]. Each map was generated at a scale of 1:50,000 and the map diagonal measures 32 km on average. Adjacent map sheets were randomly selected from the full collection, links to the original map sheets and the processed results are included with our code. Each map is stored in the GeoTIFF format [29] and is already georeferenced. The coordinates of the map corners were extracted and converted to WGS84. The images have a quality of 508 dpi (8000x10,000 pixels), which is substantially better than the other dataset. There is no additional information surrounding each raster map, so no preprocessing was required.

[7] https://www.cartesius.be/CartesiusPortal/

[8] https://www.pdok.nl/downloads/-/article/dataset-basisregistratie-topografie-brt-toprast
er

3.3.9.2 Results

The developed techniques were applied to both georeferenced datasets and the resulting predictions were compared to the ground truth geolocations. To evaluate the geolocation predictions, the mean and center georeferencing error distances were calculated. We define the mean distance as the mean haversine distance between all control points (vertices of the ground truth polygon) and the corresponding predictions. We define the center distance as the haversine distance between the predicted map center and the ground truth geolocation center. For each step of the geolocation algorithm, these error distances were calculated. For each dataset, the entire geolocation algorithm was run three times, and the results were averaged and are presented in Table 3.1.

Table 3.1: Geolocalization results for both datasets, with the average map diagonal shown in brackets. The average mean error, maximum mean error, and average center error are presented for each step of the geolocalization algorithm. Prefilter details the result from Section 3.3.4, without the final clustering step. Initial ROI denotes the result after clustering and refined ROI denotes the result after outlier filtering with RANSAC.

Dataset	Step	Mean error (km)	Max. error (km)	Center error (km)
M834 (18.94 km)	Prefilter	138	183	16.7
	Initial ROI	17.4	22.0	1.869
	Refined ROI	0.710	4.08	0.421
	Final Prediction	**0.316**	**0.631**	**0.179**
TOP50raster (32.06 km)	Prefilter	170	181	10.0
	Initial ROI	16.8	19.9	1.166
	Refined ROI	0.423	1.037	0.322
	Final Prediction	**0.287**	**0.438**	**0.162**

Max. error denotes the largest mean error for any map. Note that the center error distance does not give any indication of how large the predicted map region is compared to the ground truth geolocation and is therefore usually much smaller than the mean error. The average mean error distances for the M834 and TOP50raster datasets were 316 m (1.67% with respect to the map diagonal) and 287 m (0.90%), respectively. The largest mean error distances were 631 m (3.33%)

and 438 m (1.37%), respectively. Considering the small error distances with respect to each map's size, these results are very promising and accurate enough to use in practice. The full results for the M834 dataset are shown in Figure 3.12. Of the 16 predicted geolocations, 15 have a mean error of less than 500 m, and 11 have a center error of less than 200 m.

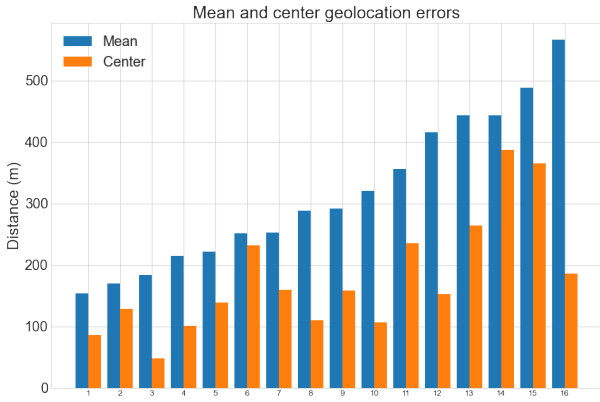


Figure 3.12: Mean and center geolocation error distances for each map in the M834 dataset.

Now that each map has a predicted geolocation, these can be imported into a GIS, using the GeoTIFF format. However, to perform any kind of analysis on the depicted geographical features, these will also need to be extracted. Most GIS offer a range of tools to semi-automatically extract relevant features of interest, but for a large collection, this is still a time-consuming process. Therefore, we performed a case study on the automatic road extraction of contemporary raster maps, using available vector data to generate the road labels. The final goal would then be to generalize this approach to historical maps, where almost no labeled data is available. Some possible approaches to perform this are described in Sections 3.6 and 6.1.

3.4 Road Segmentation

This section proposes two road segmentation approaches on a dataset of topographic raster maps. The first details a binary road segmentation approach. The second approach uses multiclass segmentation, to segment the roads by their type.

3.4.1 Dataset

For binary and multiclass road segmentation we used the same TOP50Raster dataset as in 3.3.9.1. We used the complete set of the Netherlands totaling 113 different map sheets, each with a scale of 1:50,000. Each map image measures 8,000x10,000 pixels and is in GeoTIFF format. In addition to the raster maps, the corresponding vector data is also available on PDOK. This vector dataset contains various features such as roads, waterways, buildings, tracks, etc. An example of a single map sheet is shown in Figure 3.13.

Figure 3.13: Example of one map sheet from the TOP50Raster dataset

3.4.2 Preprocessing

The vector dataset covers the entirety of the Netherlands. We used the geocoordinate bounding box of each map sheet to crop the vector dataset and selected all roads (named "Wegdeel"). We then used the GDAL library[9] to rasterize the vector data using the same resolution as the original raster map. This resulted in a binary label image representing the roads. Because these images are too large to process directly, we split each sheet into 512x512 tiles. The images and labels were zero-padded at the edges. This process resulted in a total of 31,635 images. Because the extracted roads were only a single pixel wide, these images were dilated to better match the thickness of the roads on the raster maps. After this last step, we get a result as shown in Figure 3.14. We used a 70:20:10 split for training, validation, and testing of the models in the subsequent sections.

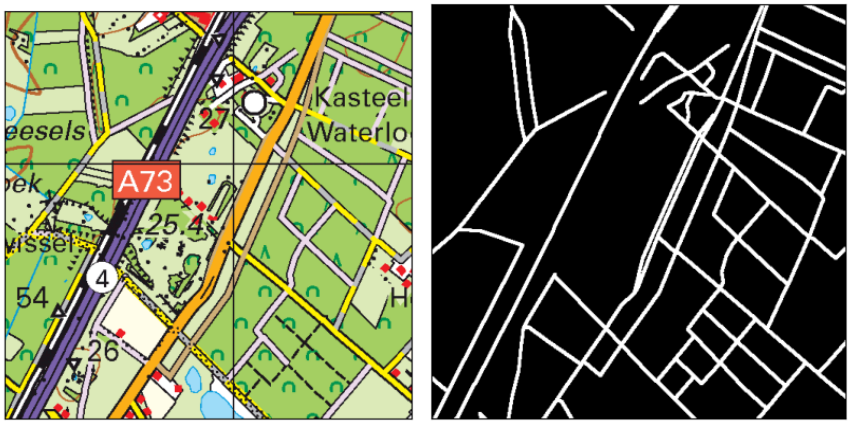


Figure 3.14: Example of one labeled tile from the dataset.

3.4.3 Binary Segmentation

For binary segmentation, we tested two model architectures, namely U-Net [30] and DeeplabV3 [31]. For each architecture, we tested four different backbones: Resnet18, ResNet50, EfficientNet-B0, and EfficientNet-B5 [32, 33]. Each backbone was initialized from weights pretrained on Imagenet. Each model was trained for a maximum of 30 epochs. We used Dice loss to train the models and saved the model with the lowest validation loss. During training, the images were augmented with random rotations and changes in contrast and brightness. Table 3.2

[9] https://gdal.org/

lists the intersection over union (IoU) and F_1 scores for each model on the test set.

Table 3.2: Binary segmentation scores

Metric	Model	ResNet18	ResNet50	EffNet-B0	EffNet-B5
IoU	U-Net	0.8029	0.8040	0.8008	**0.8043**
	DeepLabv3	0.7592	0.7699	0.7653	0.7667
F_1	U-Net	0.8605	0.8609	0.8595	**0.8618**
	DeepLabv3	0.8305	0.8388	0.8370	0.8353

The results show that the U-Net models consistently outperformed the DeeplabV3 models. The difference between backbones was small, with the largest backbone (EfficientNet-B5), achieving the best IoU score of 0.8043. The ResNet50 model was a close second, achieving an IoU score of 0.8040. When looking at the model predictions, both the U-Net and the Deeplabv3 architectures could identify most roads on the test set images. However, the predictions from the Deeplabv3 models had jagged edges and were of a lower quality. Figure 3.15 visualizes the predictions for each model on a tile from the test set.

Overall, the U-Net architecture is the most suitable for this problem. The difference in backbones is small, therefore a smaller backbone like ResNet50 or EfficientNet-B0 is preferred. The EfficientNet-B5 backbone required almost twice as much training time than the ResNet50 model. This is due to the larger number of parameters and floating point operations required to run the model.

3.4.4 Multiclass Segmentation

The PDOK vector dataset contains data for seven different road types: highway, main road, local road, regional road, street, ferry connection, and "other". These are typically represented as a different style or color on the raster maps. Because some road types are rare, we chose to create three main classes by merging the different road types. These are highway (highway class), unpaved roads ("other" class), and normal roads (remaining classes).

Using the GDAL library, we separated these respective classes and rasterized each road with a different integer value. This way, we created a multiclass segmentation dataset for each road type. The roads were dilated and each sheet was split into 512x512 tiles. The same dataset split was used as before. Based on the binary results, we trained a U-Net model using the ResNet50 backbone and dice loss, for 30 epochs. After training, we saw that the IoU score for the highway

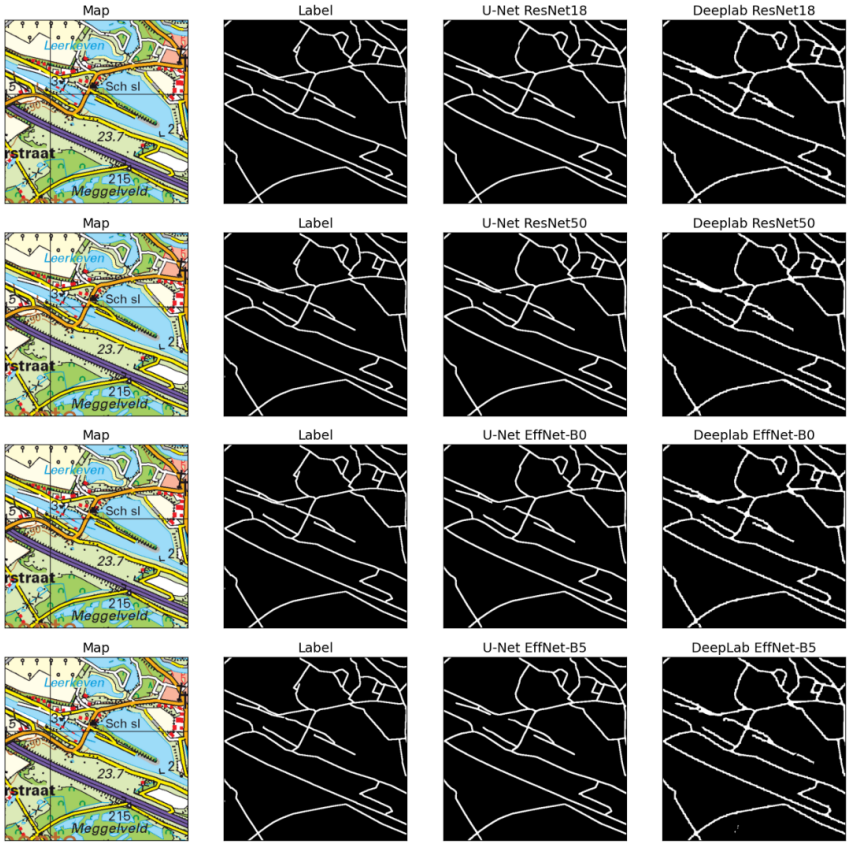


Figure 3.15: Comparison of binary segmentation model predictions.

class was zero. When looking at the pixel distribution of each class we got the following: background 94.87%, normal roads 2.93%, unpaved roads 2.09%, and highways 0.11%. We then retrained the model using Focal loss [34], which has a better performance on large class imbalances. After this change, the model correctly predicted the highway class. Table 3.3 lists the IoU scores for each road type and loss function. Figure 3.16 shows the model predictions on a tile from the test set.

While the results are less accurate than the binary segmentation, these are still good. We suspect that these scores can still be improved by using additional augmentation methods and perhaps by restructuring the classes. We noticed that sometimes there was a small change of road type in the vector data, but this was not reflected in the raster map. Such cases may harm the model's performance. To integrate these results into a GIS, the road masks still need to be vectorized.

Table 3.3: Overview of the IoU scores per road type and loss function.

Loss	Highway	Road	Unpaved	Average
Focal Loss	**0.736**	**0.791**	**0.748**	**0.758**
Dice Loss	0.000	0.714	0.664	0.459

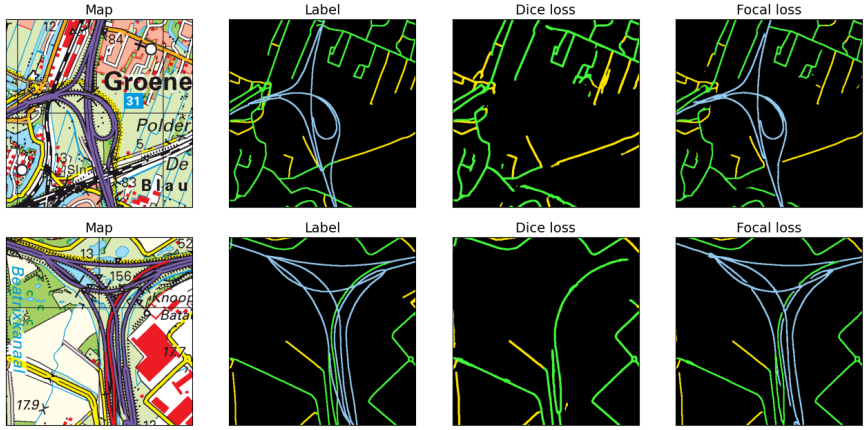


Figure 3.16: Comparison of loss functions for multiclass segmentation. Highways were not detected using the Dice loss function.

3.5 Case study: Walking Route Segmentation

This section details a case study on the automated processing of walking and cycling maps, where the goal was to estimate GPS coordinates for the highlighted route. This case study was part of a collaborative project with RouteYou [10].

To extract the highlighted walking or cycling route from a given map image, we constructed an automated processing pipeline based on our developed methods from Sections 3.3 and 3.4. First, the maps were converted to images from PDF, where we saved each page as an image. Next, we used a YOLOv5 object detection model to locate and crop the map(s) on each image. After cropping, we performed text detection and geolocalization using the same methods as in 3.3. A custom color-based method was used to generate labels for the highlighted routes. These labels were then used to train a custom U-Net model. Finally, the route was post-processed and matched with the underlying street network via OpenStreetMap (OSM).

[10] https://www.routeyou.com/

The walking and cycling routes were gathered from two sources. We used a collection of 132 walking and 115 cycling maps from Tourisme Oost-Vlaanderen (TOV) [11]. Each of the maps was depicted on a single image and contained a single GPX (GPS Exchange Format) file with the route coordinates. The second dataset consisted of 656 walking and cycling maps from France gathered via web scraping. These PDFs often contained multiple maps and routes. The combined dataset contained a total of 1987 images after splitting the PDFs per page.

3.5.1 Route Segmentation

To crop the maps from the full image, we used an edge detection approach combined with horizontal line detection via morphological operations. Figure 3.17 shows an example image and the mask of the largest connected component of the edges that was used to crop out the map. To segment the highlighted route from each map, we first tried a color-based approach via trial and error on the 247 maps from the TOV dataset.

Figure 3.17: Left: Original image of the route, with the detected map region highlighted. Right: Largest connected component after edge detection and dilation.

Each map typically only contained a few colors and the actual route was often centrally placed in a unique color. Via k-means clustering, we reduced the number

[11] https://www.routen.be/

of colors in each image to 16. Each road on the map was always represented as a line with a certain thickness. This thickness was much smaller than the filled polygons on the map that represented the different types of terrain (grass, industry, water, etc.). Therefore, we used the distance transform to classify the road colors. The distance transform produces another image where the value of each pixel represents the distance to the nearest zero pixel (background pixel) in the input mask. Therefore, the colors depicting roads had a much lower maximum value of their distance transform. We could then easily separate the road colors, by clustering the maximum distances for each color mask. We subsequently merged all these road colors into a single mask that represented all of the roads.

While this mask typically contained all the roads, it also contained text and other symbols that overlapped with the roads or were represented in the same color. Therefore, we used the ground truth route coordinates to filter out the actual route. We rasterized these coordinates, and performed template matching to match the route with the mask containing all roads. Because the geocoordinates were scaled differently than the image, we iteratively rescaled the width and height independently from 0.5 to 1 and saved the parameters with the highest matching score. Finally, we can use an AND operation to get the correct mask of the route. To speed up this algorithm, the images were downscaled by a factor of four. Figure 3.18 shows the mask of all detected roads, the best template match, and the resulting final mask.

This approach worked well on the dataset of TOV, but performed quite poorly on the dataset of French maps, even after cropping out the maps as described in 3.5.2. Therefore, we used a simple interactive script to overlay the predicted route on top of the map. Then, using keyboard inputs, these predictions were quickly accepted or rejected. Ultimately, only 279 out of 903 (30.1%) of the predicted routes were accurate. We then used these correct predictions to train a U-Net segmentation model and selected 32 random images for validation. The model achieved an IoU of only 0.266 on the validation set. Figure 3.19 shows the predictions on some maps from the validation set. The low IoU score is mainly due to the difference in thickness of the predicted versus the actual route and due to non-route roads being included in the predictions. Nevertheless, most route predictions include the actual route. Therefore, with some post-processing, these can be improved.

First, we performed k-means clustering on the original map image. Then, we used the U-Net prediction to select the top 3 colors with the most overlap with the prediction. Next, these color masks were filtered if they touched the map edges, as most routes were placed in the middle of the map. We then applied skeletonization to get a mask of one pixel wide. Sometimes, the resulting route had holes or loops. We simply connected each edge with its closest unconnected edge using a straight line, to fill the gaps. Then, we represented the mask as a

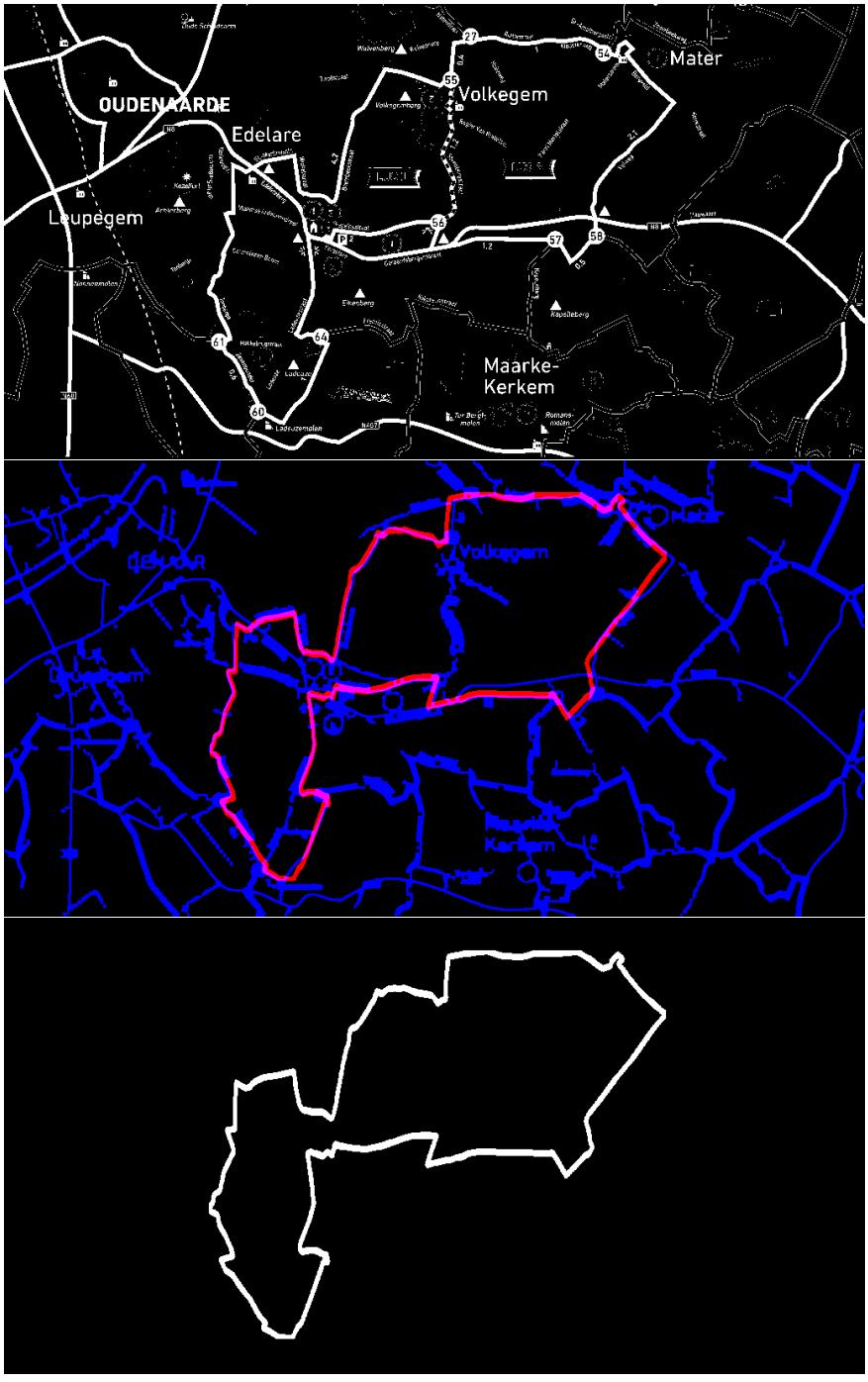


Figure 3.18: From top to bottom: Mask of all detected road colors. Best template match (red) overlaid on all roads (blue). Final mask rescaled to the original image dimensions.

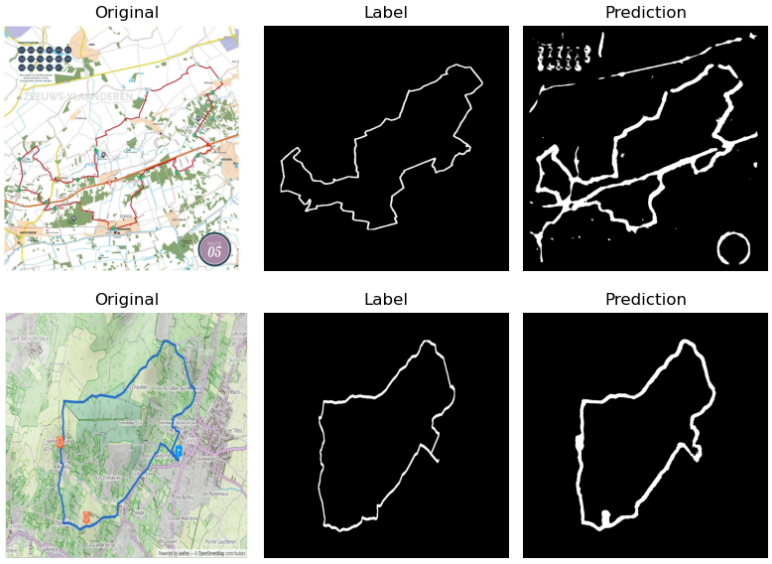


Figure 3.19: Walking route predictions of the U-Net model on two maps from the validation set.

graph and used NetworkX [12] to find the longest non-looping path. This path was then selected as the final route prediction.

3.5.2 Map Detection

To detect the map(s) on each image from the French dataset, we used a YOLOv5 object detection model. We manually labeled all of the images from the French dataset with each map's bounding box. 128 images were randomly selected for validation. We trained a YOLOv5 object detection model, which achieved a mean average precision (mAP) of 0.89 on the validation set. Figure 3.20 visualizes the map predictions on an image from the validation set.

3.5.3 Route Geolocalization

Now that the maps have been detected and the route has been segmented, we can assign geocoordinates to the route. Using the same approach as in 3.3, we estimated a geolocation for each map. For the TOV dataset, we calculated the

[12] https://networkx.org/

Figure 3.20: Map detection predictions via YOLOv5 on an image from the validation set.

geolocation errors, as each map was linked to a single route. For the dataset of French maps, these were not linked to a single route, so they were difficult to validate.

The geolocation error was estimated by taking the bounding box of the ground truth GPS route and comparing this to the predicted geolocation. Keep in mind that the predicted geolocation was a prediction for the full map, so these regions were typically larger than the ground truth route. Nevertheless, we achieved a mean geolocation error of 2981m and 4940m for the walking and cycling routes, respectively. The average center errors were 843m and 1211m, respectively (see 3.3.9.2 for an explanation of these metrics).

As a final step, we used the predicted geolocation to assign geocoordinates to the predicted route. Next, we used a map-matching algorithm to match our predictions with the underlying road network via OSM. Figure 3.21 shows one of the better results after geolocating and map matching. The final route predictions were often far away from the actual route, as they depended on both the route and

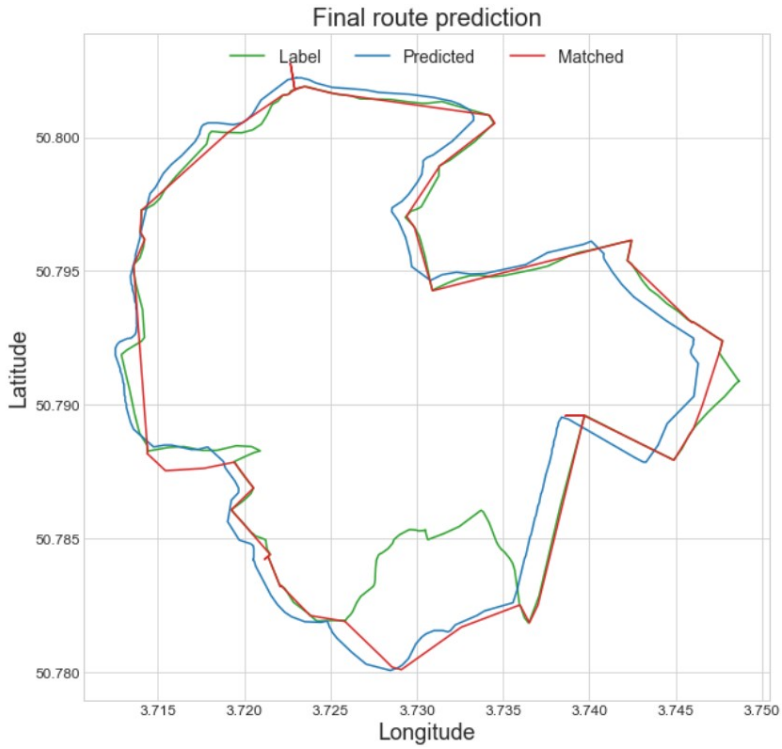


Figure 3.21: Final route prediction (blue) with the ground truth (green) and matched (red) routes visualized.

geolocation to be predicted accurately. This was especially true for the dataset of French maps, which were more complex and typically contained fewer toponyms.

3.6 Discussion

We developed an automatic map processing approach that can extract visible toponyms on raster maps and subsequently georeference these maps, with a relatively small error. It is surprising that the predictions for the TOP50raster dataset, which contains much larger maps, were better than the other dataset. There are two main reasons for this. First, the maps of TOP50raster are of a higher quality (508 dpi versus 225 dpi), which should lead to better text detection and recognition performance. Second, the average number of detected text labels is nearly twice as high, while the distribution of the number of geocoder matches per text label is similar (see Figure 3.6). This results in the geolocation algorithm receiv-

ing much more valuable information on average, resulting in a better prediction. Detecting more text labels improves the predictions only if they can be matched with the corresponding toponym.

Table 3.1 shows the improvements for each step in the geolocation algorithm. It is clear that simply clustering the initial geocoder matches gives inaccurate results compared to the final prediction. The error distances also show that the center error is not a good indicator of overall accuracy. As we only compared the predicted and ground truth center, there is no indication of the relative scale of the predicted area. Without filtering, the geolocation algorithm consistently predicted much larger areas than the actual map.

For many applications, an average error of less than 2% is usable. These maps can now be integrated into a GIS with the recognized text labels and matched toponyms as additional metadata. Besides the raster maps themselves, we only used the country information. This was provided to the geocoders to reduce the number of incorrect toponym matches. For most collections, it is trivial to provide the country information to the algorithm. For maps depicting border regions of countries, it can be beneficial to include both countries in the geocoder requests. Besides querying the recognized text labels with the geocoders, the entire processing pipeline, from text detection and recognition to geolocation prediction is relatively fast. The calculations necessary to perform the outlier filtering and to calculate the error metric presented in Section 3.3.7 were nearly all vectorized and are therefore very performant.

However, it remains difficult to automatically determine if the georeferencing predictions were accurate. This is a classic problem that plagues many unsupervised approaches, as no labeled data would be available when using this technique in practice. We can, however, perform some extra validations, given that the raster map is part of a uniform series (which is usually the case for topographic maps). For instance, if the height or width of the predicted geolocation for a map is much greater than the other predictions, we can assume that this prediction is less accurate.

The backbone of this pipeline is the determination of the initial region of interest. Because density-based clustering was used, this determination assumes that the text recognition was usable and that the number of false-positive geocoder matches was relatively small compared to the number of correct matches. Therefore, the two main failure cases of the geolocation approach occur when the map is of low quality, resulting in poor text recognition results, or when the text labels are consistently split into multiple, far away words or presented in complex arrangements. Text recognition techniques that can work with low-quality images, such as [35], could be used to improve results. But usually, this is not an issue when analyzing raster maps. The main issue is often the correct linking of text labels with the same location phrase. This was not the focus of our research and

is not a trivial task to solve, as the impressive work in [17] shows. Even with a complex technique to correctly link the text labels, errors remained. In that study, only the center of each map was geolocated. The center of the largest cluster of geocoder matches was taken as a prediction for the map center, which resulted in errors of 27%, 48%, and 51% with the ground truth center, relative to the map diagonals, for the three maps discussed in detail. Without any text linking, these errors were over 91%, clearly demonstrating its importance in predicting an initial region of interest.

Even though we are satisfied with the results presented, we believe that there is still much room for improvement. Mainly the text linking and geocoder querying need improvement, as these are key in predicting a correct initial region of interest. Adding additional semantic information can also help determine more robust geolocations. Currently, each text label and corresponding toponym is naively considered as a point. These toponyms often do not represent single points, but lines and polygons. Clearly, this is not the optimal way to deal with these types of features. Additionally, text labels that denote visible features on the map, such as rivers and streets could be used in conjunction with the visual content to more accurately geolocate these. It can also be beneficial to give different features a higher weight, depending on their type. For instance, a street name may provide more useful and localized information than a place name. Finally, more work needs to be done on developing an effective strategy to validate the predicted geolocation in an unsupervised way. It can be valuable to know if the predictions were not accurate for certain maps in the dataset; these can then be corrected manually.

Furthermore, we showed how U-Net models can effectively segment the roads on contemporary topographic maps. We were able to construct a multi-class dataset of different road types, by linking the raster maps with associated vector data. However, for historical maps, no such vector data typically exists. Synthetic data generation or domain adaptation techniques, such as [22, 23], have to be used to train the models in a weakly-supervised way.

Our case study on walking and cycling maps showed another use case for the geolocation and segmentation approaches. By combining these, we were able to predict GPS coordinates for the highlighted routes on each map. While the full pipeline did not always work perfectly, this is (to the best of our knowledge) the first automatic pipeline that can achieve this feat.

3.7 Conclusion

We have developed an automatic technique to georeference topographic raster maps using pretrained text recognition models and geocoders. Two datasets were

processed, resulting in an average error of 316 m (1.67%) and 287 m (0.90%) for maps spanning 19 km and 32 km, respectively. With average errors within 2% of the map size, these maps can now be accurately queried for a specific region of interest. The georeferenced maps can then be integrated into a GIS with the recognized text labels and linked open data toponyms as additional metadata for each raster map in the collection. This additional metadata greatly improves the quality and accessibility of the dataset. Furthermore, we showed how U-Net models can effectively segment the roads on contemporary topographic maps, by linking the raster maps with associated vector data. Our ResNet50 segmentation model achieved an IoU score of 0.804 and 0.758 on the binary and multi-class datasets, respectively. We then combined both methods, to propose a novel walking route prediction pipeline, that produces GPS coordinates from a given map image.

References

[1] V. Crăciunescu, Ş. Constantinescu, I. Ovejanu, and I. Rus. *Project eHarta: a collaborative initiative to digitally preserve and freely share old cartographic documents in Romania*. e-Perimetron, 6(4):261–269, 2011.

[2] C. Fleet, K. C. Kowal, and P. Pridal. *Georeferencer: Crowdsourced georeferencing for map library collections*. D-Lib magazine, 18(11/12), 2012.

[3] T. Waters. *Map Warper*, 2020. https://mapwarper.net/.

[4] A. Pezeshk and R. L. Tutwiler. *Automatic feature extraction and text recognition from scanned topographic maps*. IEEE Transactions on Geoscience and Remote Sensing, 49(12):5047–5063, 2011.

[5] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. *Photoocr: Reading text in uncontrolled conditions*. In Proceedings of the IEEE International Conference on Computer Vision, pages 785–792, 2013.

[6] Y.-Y. Chiang and C. A. Knoblock. *A general approach for extracting road vector data from raster maps*. International Journal on Document Analysis and Recognition (IJDAR), 16:55–81, 2013.

[7] A. P. Tafti, A. Baghaie, M. Assefi, H. R. Arabnia, Z. Yu, and P. Peissig. *OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym*. In International Symposium on Visual Computing, pages 735–746. Springer, 2016.

[8] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. *ICDAR 2015 competition on robust reading*. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 1156–1160. IEEE, 2015.

[9] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. *Coco-text: Dataset and benchmark for text detection and recognition in natural images*. arXiv preprint arXiv:1601.07140, 2016.

[10] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. *What is wrong with scene text recognition model comparisons? dataset and model analysis*. In Proceedings of the IEEE International Conference on Computer Vision, pages 4715–4723, 2019.

[11] K. Milleville, S. Verstockt, and N. Van de Weghe. *Improving toponym recognition accuracy of historical topographic maps*. In International workshop on Automatic Vectorisation of Historical Maps. ELTE Eötvös Loránd University. Department of Cartography and Geoinformatics, 2020.

[12] S. Abkenar and A. Ahmadyfard. *Text Extraction from Raster Maps Using Color Space Quantization*. pages 77–86, 01 2017. doi:10.5121/csit.2017.70208.

[13] W. Höhn. *Detecting arbitrarily oriented text labels in early maps*. In Iberian Conference on Pattern Recognition and Image Analysis, pages 424–432. Springer, 2013.

[14] Y.-Y. Chiang and C. A. Knoblock. *Recognizing text in raster maps*. Geoinformatica, 19(1):1–27, 2015.

[15] J. Weinman. *Geographic and style models for historical map alignment and toponym recognition*. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 1, pages 957–964. IEEE, 2017.

[16] S. Tavakkol, Y.-Y. Chiang, T. Waters, F. Han, K. Prasad, and R. Kiveris. *Kartta labs: Unrendering historical maps*. In Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, pages 48–51, 2019.

[17] Z. Li, Y.-Y. Chiang, S. Tavakkol, B. Shbita, J. H. Uhl, S. Leyk, and C. A. Knoblock. *An Automatic Approach for Generating Rich, Linked Geo-Metadata from Historical Map Images*. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 3290–3298, 2020.

[18] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. *Character region awareness for text detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9365–9374, 2019.

[19] B. Shi, X. Bai, and C. Yao. *An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11):2298–2304, 2017. doi:10.1109/TPAMI.2016.2646371.

[20] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. *East: an efficient and accurate scene text detector*. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 5551–5560, 2017.

[21] C. Jiao, M. Heitzler, and L. Hurni. *A survey of road feature extraction methods from raster maps*. Transactions in GIS, 25(6):2734–2763, 2021.

[22] W. Duan, Y.-Y. Chiang, S. Leyk, J. H. Uhl, and C. A. Knoblock. *Automatic alignment of contemporary vector data and georeferenced historical maps using reinforcement learning*. International Journal of Geographical Information Science, 34(4):824–849, 2020. Available from: https://doi.org/10.1080/13658816.2019.1698742, arXiv:https://doi.org/10.1080/13658816.2019.1698742, doi:10.1080/13658816.2019.1698742.

[23] S. Wu, K. Schindler, M. Heitzler, and L. Hurni. *Domain adaptation in segmenting historical maps: A weakly supervised approach through spatial co-occurrence*. ISPRS Journal of Photogrammetry and Remote Sensing, 197:199–211, 2023. Available from: https://www.sciencedirect.com/science/article/pii/S0924271623000278, doi:https://doi.org/10.1016/j.isprsjprs.2023.01.021.

[24] J.-P. Donnay and P. Lambot. *Geodetic and cartographical standards applied in Belgium*. A Concise Geography of Belgium, pages 41–42, 2012.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, page 226–231. AAAI Press, 1996.

[26] M. A. Fischler and R. C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM, 24(6):381–395, 1981.

[27] P. De Maeyer, B. M. De Vliegher, and M. Brondeel. *Spiegel van de wereld*. Academia Press, 2004.

[28] J. E. Stoter, M.-J. Kraak, and R. Knippers. *Generalization of framework data: A research agenda*. In ICA Workshop on "Generalisation and Multiple representation, pages 20–21, 2004.

[29] N. Ritter and M. Ruth. *The GeoTiff data interchange standard for raster geographic images*. International Journal of Remote Sensing, 18(7):1637–1647, 1997.

[30] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241, Cham, 2015. Springer International Publishing.

[31] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, Computer Vision – ECCV 2018, pages 833–851, Cham, 2018. Springer International Publishing.

[32] K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[33] M. Tan and Q. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. In K. Chaudhuri and R. Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. Available from: https://proceedings.mlr.press/v97/tan19a.html.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. *Focal loss for dense object detection*. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[35] W. Wang, E. Xie, P. Sun, W. Wang, L. Tian, C. Shen, and P. Luo. *TextSR: Content-aware text super-resolution guided by recognition*. arXiv preprint arXiv:1909.07113, 2019.

4

Analyzing Herbarium Sheets

"Study the past, if you would divine the future"

– Confucius

In this chapter, automated processing methods for digitized herbarium sheets are discussed. The chapter starts with an overview of the challenges related to herbarium processing, followed by related work. Next, automated processing methods are discussed, from preprocessing to specimen identification via OCR. The chapter finishes by proposing and evaluating multiple herbarium segmentation approaches on a novel dataset.

This chapter is an adapted version of the following publication:

Milleville, K., Thirukokaranam Chandrasekar, K. K., Van de Weghe, N., & Verstockt, S. (2023). **Evaluating Segmentation Approaches on Digitized Herbarium Specimens**. In: Bebis, G., et al. Advances in Visual Computing. ISVC 2023. Lecture Notes in Computer Science, vol 14362. Springer, Cham. https://doi.org/10.1007/978-3-031-47966-3_6

Which improves upon our earlier work:

Milleville, K., Thirukokaranam Chandrasekar, K. K., & Verstockt, S. (2023). **Auto-**

matic Extraction of Specimens from Multi-specimen Herbaria. ACM Journal on Computing and Cultural Heritage, 16(1), 1-15. https://doi.org/10.1145/3575862

4.1 Introduction

Herbarium specimens record plant occurrences collected from all corners of the world, forming the foundation of systematic botany. They have been collected over several centuries and are carefully archived and preserved. Each specimen, typically a dried plant, is attached to a herbarium sheet. These sheets also contain essential information such as the plant's scientific name, collection date, geographical origin, and other relevant details. The herbarium sheets thus form a physical database of plant biodiversity and are used to study species diversity and their evolution over time. Furthermore, herbaria offer a unique opportunity to study past ecological conditions and the effects of climatic and other changes on plant populations. Following a report by the Index Herbatorium from 2021, there are close to 400 million herbarium specimens spread over 182 countries [1].

In the last few decades, many institutions have begun digitizing their archives and making their specimens accessible on various online repositories like GBIF[1] and iDigBio[2]. This digitization effort has led to a dramatic increase in accessibility and research focused on herbarium specimens. The digitization process involves taking a photograph or making a scan of the specimen followed by (manual) data entry for the specimen details (location, date, taxonomy, collector, etc.) [2]. This digitization is time-consuming, considering many collections contain thousands or millions of herbarium sheets. Usually, a ruler and color card are added during digitization to provide color and size references, that are useful for later analysis.

Most herbarium sheets contain little or no metadata about the size and shape of the specimens. Therefore, performing large-scale studies on the morphological features of these specimens typically requires a tremendous manual effort. Several semi-automated tools exist, but most still require manual annotation or corrections, which limits the scope of such studies [3, 4]. Luckily, computer vision and deep learning methods can be used to automatically analyze digitized herbarium specimens. The resulting metadata can then be used to enrich the digitized specimens, which allows researchers and botanists to delve deeper into plant biodiversity studies.

[1] https://www.gbif.org/

[2] https://www.idigbio.org/

4.2 Related Work

In recent years, deep learning-based object detection and segmentation models have become the most popular state-of-the-art methods for analyzing digitized herbarium sheets. Object detection approaches work well for clearly defined objects and are easy to label. However, plant specimens can have a complex, non-convex shape, which makes these bounding boxes inaccurate with regard to the actual shape. Therefore, image segmentation techniques are preferred. Different segmentation models have been developed over the years to tackle these tasks. For semantic segmentation, U-Net [5] has been widely used, especially in the biomedical field. DeepLabV3+ [6] is another notable model used for semantic segmentation tasks. For object detection, YOLO [7] models are popular and performant. YOLOv8 [8] is one of the newer YOLO architectures, that is also capable of instance segmentation. Detectron2 [9] is another popular framework for instance and panoptic segmentation. It includes implementations of several popular models, such as Mask R-CNN [10]. More recently, vision transformers, which typically contain both convolutional and transformer layers, are among the state-of-the-art for segmentation tasks. Mask2Former [11] and OneFormer [12] are notable examples, which achieve outstanding accuracy and combine the three segmentation tasks into a single unified model. However, these newer architectures typically require more computational resources.

Semantic segmentation methods are frequently used to separate the specimen(s) from the background and tend to be very accurate. In [13], the authors published a dataset of 400 digitized fern specimens. They used a color thresholding method with manual corrections to extract the specimens from the background. Their retrained U-Net model achieved an F_1 score of 0.95. Similarly, [14] retrained DeeplabV3 and FRRN-A [15] models on a custom dataset of 395 herbarium specimens, achieving mIoU (mean intersection over union) scores of 0.981 and 0.992, respectively. Similar approaches are also applied to other types of specimens. In [16], the Mothra toolkit was developed to segment moth specimens, labels, and rulers, which were used to measure phenotypic characteristics. They then applied this method to over 180,000 specimens to perform large-scale studies.

The detection of leaves or other objects is typically performed via object detection or instance segmentation models. In [17], a modified YOLOv3 model was used to detect plant organs (leaves, buds, flowers, and fruits). After data augmentation, they achieved an F_1 score of 0.938, compared to 0.899 without augmentation. Deep Leaf [4] is an instance segmentation model, based on Mask R-CNN, that segmented leaves and common objects from herbarium specimens. It achieved a mIoU of 0.905 for the leaf segmentation on a dataset of 4000 images. Furthermore, the length of the recognized rulers was used to accurately estimate leaf

morphological traits.

4.3 Digitization and Preprocessing

In collaboration with the Meise Botanic Garden, we assisted the Ghent University archives with their ongoing digitization process. The digitization process involves photographing each herbaria sheet, identified by a unique barcode. A standardized color card is placed next to each sheet, to provide color and size references. We assisted this digitization effort by automatically extracting the herbarium sheets and color cards, which were then stitched together. Figure 4.1 shows one of the digitized sheets.

Figure 4.1: Example of a digitized herbarium sheet from the collection.

4.3.1 Page Extraction

The preprocessing pipeline begins with the extraction of the page from the image and is adapted from the methods developed in [18, 19]. First, the image is rotated such that the longest edge is maintained as its height. Next, the page is masked

from the image using a threshold of 136 on the grayscale values. This masked the page from the darker background. Using this mask, the largest connected component was selected as the page. Finally, the contour of the page mask was used to dewarp the page into a straight rectangle. By selecting 32 points along the contour, the image contents are remapped and interpolated into the correct shape using the OpenCV remap function[3]. Figure 4.2 shows the extracted page mask and selected points along the page edge before dewarping.

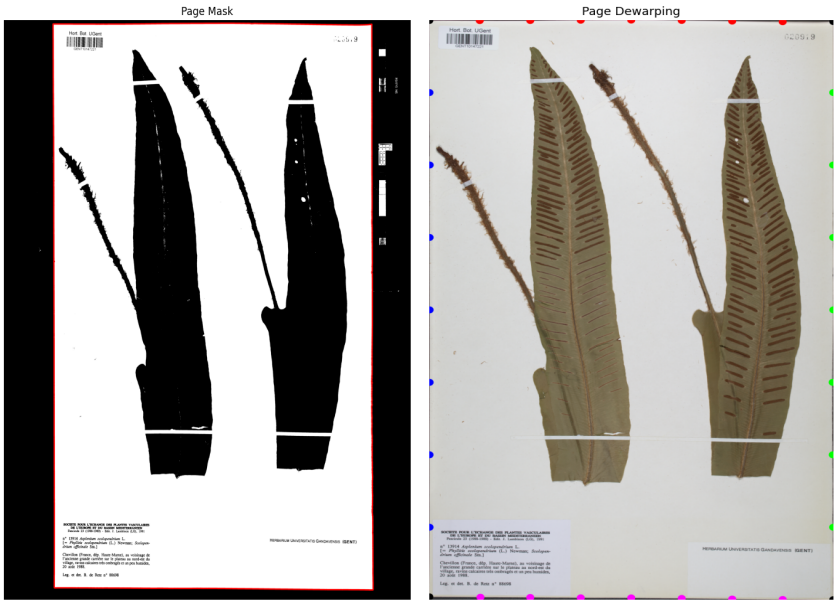


Figure 4.2: Left: page mask and its contour in red. Right: selected points along the page edge before dewarping.

4.3.2 Color Card Extraction

To extract the color card from each image, we used a manually cropped color card as a template for object detection. Because the same color card is always used during digitization, there is no need to train an object detection model. Classic feature extraction and matching techniques can be used effectively. After some initial testing, we found that ORB features [20] worked well to detect the color card. The position of the card was then determined using the bounding box of the matched features and also via a homography approach using RANSAC. We found

[3] https://docs.opencv.org/4.7.0/d1/da0/tutorial_remap.html

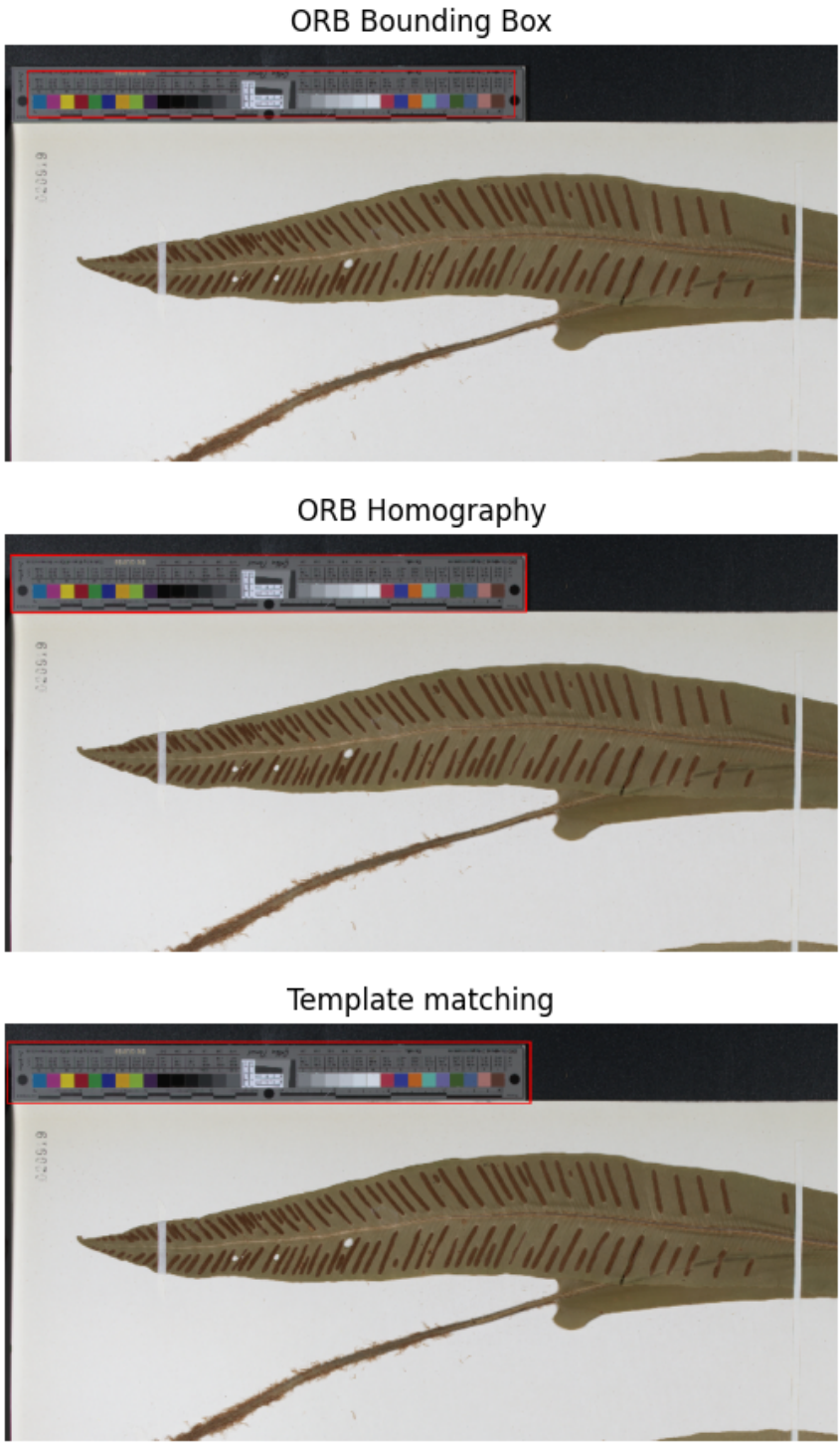


Figure 4.3: Results of the three color card detection methods.

that the simple bounding box approach was consistently inaccurate, therefore the homography approach was used. This approach did not always work perfectly, so if the detection failed, we used a template matching technique. While template matching was consistent, the downside was that only the top-left corner of the card was found. If the card was slightly rotated in the photograph, this was not detected, resulting in an inaccurate bounding box. Figure 4.3 shows the three detections generated via each method.

Figure 4.4: Final result of the preprocessing pipeline.

After segmenting both the page and color card, these were stitched together and the Ghent University logo was added. The resulting image was also converted to lossless TIFF for further processing. Figure 4.4 shows the final result. In an initial test using 3000 digitized herbarium sheets, we noticed that the above detec-

tion methods were often slightly inaccurate in detecting the color cards. Therefore, a small change was made during the digitization process. The color card was placed on top of a bright red piece of paper, which completely surrounded it. This way, the red color could easily be detected along with the color card. We manually validated 3507 images using this new method and found that 3423 (97.6%) were preprocessed correctly. The errors were usually due to incorrect dewarping or page extraction when the sheets had an irregular shape.

4.4 Specimen identification

This section details the use of Azure OCR[4], to recognize and match a specimen's species and genus, from the herbarium text labels. The genus and species are the lowest levels of the plant taxonomy classification, thus providing the most specific information. This information is typically already available in a digital format, or it has to be added manually during digitization. If this information was already available, the automatic identification could be used to validate the digitization.

A test was performed using a random sample of 400 digitized herbarium specimens from the LifeCLEF 2020 Plant Identification Challenge [21]. These specimen images are accompanied by XML files, detailing their genus, species, and other additional metadata. After a sanity check of these XML files, we found that some could not be correctly parsed, resulting in 369 usable files containing both the genus and species.

The OCR results were consistently accurate for printed text, but are typically less accurate for handwritten text. Almost every specimen was accompanied by one or more printed text labels, so this was not an issue for this test. Figure 4.5 shows the output of Azure OCR for one of the specimens. Many specimens also contain a barcode, which can either be extracted via OCR or via a barcode reader like Zebra Crossing[5], providing an additional validation step during digitization.

The OCR results are structured as lines, that each contain a list of recognized words. If a line is longer than six words, it is ignored because it is unlikely to contain the plant's name. The names of the plants are usually written on a separate line with sometimes one word in front of it, such as "name" or "genus". Other times, the canonical name of the plant is followed by an abbreviation like "L." or "O.", this typically indicates the name of the botanist credited for the plant name.

Lines containing less than three characters or consisting solely of numbers or punctuation marks are discarded. The list of lines is then trimmed down to the first two words. These two words usually contain the full canonical name of

[4] https://azure.microsoft.com/en-us/products/ai-services/ai-vision

[5] https://github.com/zxing-cpp/zxing-cpp

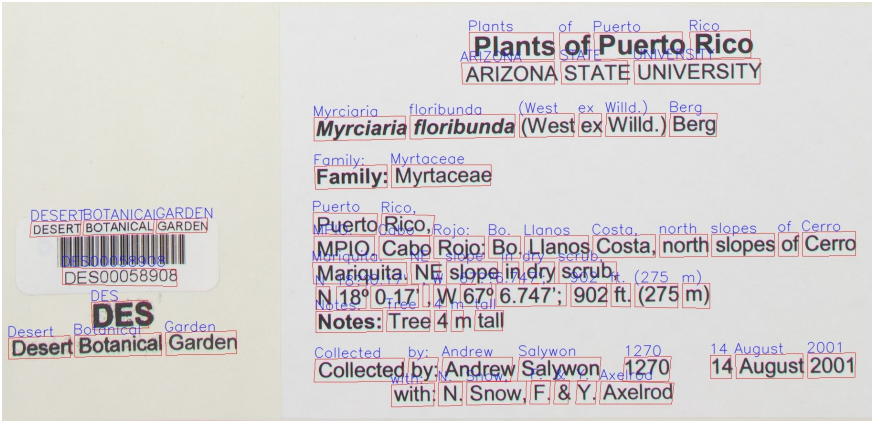


Figure 4.5: Output of Azure OCR on one of the specimen labels.

the plant. Words that are shorter than four letters or contain only numbers or punctuation marks are discarded. Finally, all words are converted to lowercase to reduce matching errors.

4.4.1 Text Matching

To match the recognized words as a correct genus or species, the GBIF backbone taxonomy [22] was used. This database contains the canonical name, genus, and species. Because the OCR results still contain some errors, we cannot fully rely on exact string matching. Sometimes visually similar letters or groups of them are recognized as different letters. For instance, "cl" could be recognized as "d" or "g" could be recognized as "q". Therefore, a fuzzy matching approach was used, using the popular TheFuzz library[6]. This library offers multiple fuzzy string matching functions, based on the Levenshtein distance.

As a first step, an exact match is sought for the extracted word pairs. This string comparison was significantly faster than fuzzy matching and almost always resulted in a correct match. If no exact match was found, the algorithm switched to matching separate words. The algorithm will try to match the separate words with a genus or species from the database. Once a list has been compiled with these exact matches, a search was conducted for each genus for an applicable species and vice versa.

If none of these combinations was a correct plant species, the algorithm switched to fuzzy matching. The lists of possible matches were first narrowed

[6] https://github.com/seatgeek/TheFuzz

down so that only applicable species and genera were considered based on the previously found part of the plant name. When a word matched with a score of 80 or more, it was saved as a possible candidate for the genus or species of the plant until a better match was found. The match with the largest score was ultimately considered the chosen plant name.

It is also possible to look for a fuzzy match for both the genus and the plant species, but this step significantly slows down the algorithm. The algorithm typically took around five minutes to complete. When this additional fuzzy matching step was added, it ran up to fifty times slower. Besides this massive speed decrease, the precision of this step of the algorithm fell to a little over 5 percent. With this level of precision, every specimen would have to be checked manually, which defeats the purpose of automatic identification.

4.4.2 Results

When the algorithm found a match for a plant name, it also indicated how the genus and species were identified. The algorithm distinguishes between three possibilities:

- The name was found by an exact match of both parts (Exact).
- The name was found by a fuzzy match of the genus (FuzzG).
- The name was found by a fuzzy match of the species (FuzzS).

TheFuzz has six different ways to calculate the similarity between two strings. After testing each similarity score, we found that the Token Sort Ratio produced the best results. This ratio was then used during the evaluation.

When an exact match of both genus and species was found, the precision was 1, and the recall was 0.39. Using the full algorithm with fuzzy matching, the precision dropped to 0.72 for the genus and 0.66 for the species. The recall, however, greatly increased to 0.75 and 0.74, respectively.

When looking at incorrect matches, we identified three major causes. Firstly, there are cases where the XML files deviate from the name present on the image. Secondly, there are a large number of cases where a place name mentioned on the page is interpreted as the species of the plant. This is often due to the words "Paris" or "flora". "Paris" appeared many times due to a large number of herbarium sheets originating from Paris and flora is also frequently present on the text labels. The last cause lies in the text recognition errors of the OCR. Table 4.1 shows the precision and recall for the full algorithm and how names were matched, with and without the inclusion of the words "Paris" and "flora". By excluding these common words, the precision increased for both species and genera,

while the recall decreased. While these fuzzy results are okay, their precision is likely too low to be used effectively as a validation tool.

Table 4.1: Precision and recall for the full algorithm and how names were matched. Tests were performed with and without inclusion of the words "paris" and "flora". All tests were performed using the Token Sort Ratio.

Genus		
Algorithm	All	Without Paris & flora
Full	0.72 | 0.75	0.81 | 0.69
Exact	1.00 | 0.39	1.00 | 0.39
FuzzG	0.66 | 0.63	0.71 | 0.59
FuzzS	0.75 | 0.56	0.92 | 0.51
Species		
Algorithm	All	Without Paris & flora
Full	0.66 | 0.74	0.74 | 0.67
Exact	1.00 | 0.39	1.00 | 0.39
FuzzG	0.68 | 0.64	0.74 | 0.59
FuzzS	0.60 | 0.50	0.71 | 0.44

4.5 Herbarium Specimen Segmentation

This section details the different segmentation approaches used to analyze the herbarium sheets. These methods improve upon our previous work [18] and are evaluated on a novel instance segmentation dataset. First, binary segmentation models are fine-tuned to segment the specimens from the background. Next, several instance segmentation models are compared. Finally, a comparison is made between combining both types of models and using a single panoptic segmentation model to segment both specimens and objects.

4.5.1 Dataset

The herbarium specimen dataset was created using a semi-automatic approach. The labeling was split into two parts, namely, the semi-automatic labeling of the

plant specimen(s) followed by the manual annotation of common herbaria objects. We started with a random sample of 1500 images from the LifeCLEF 2020 Plant Identification Challenge [21] and also included the 250 specimens from [23]. These images were resized such that their longest side measured 1024 pixels. We used this dataset to fine-tune our semi-automatic labeling approach for the plants.

First, the images were transformed into the LAB color space, which groups similar colors closer together than the traditional RGB color space. Next, k-means clustering was used with a k value of 3, to reduce the number of colors in the image. This facilitated the extraction of the foreground regions (plants and objects) from the background sheet. Then, the foreground regions were split via a connected components analysis.

We noticed that most non-specimen objects were rectangular and close to the image borders. Therefore, these were filtered based on their relative size, shape, and overlap with the page borders. For each foreground object, we calculated the overlap with the page border (outer 10% of the image) and the overlap of its area compared to the area of its bounding box. If these exceeded a predefined threshold, the object was filtered out. These thresholds were determined qualitatively and the full algorithm details are made available with our code. After filtering, each remaining specimen's mask was saved independently. Figure 4.6 shows the result after extracting the foreground objects and after filtering the non-specimen objects.

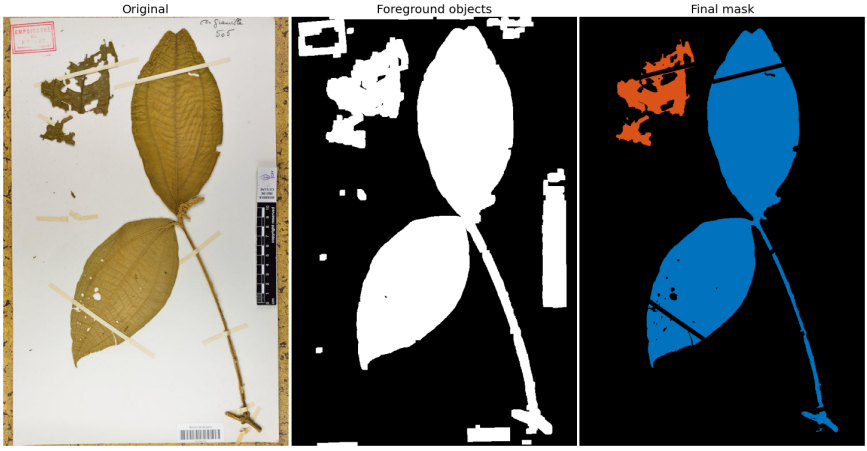


Figure 4.6: From left to right: The original herbarium image. Result after segmenting and dilating the foreground objects. Final result after filtering the non-specimen objects (each color denotes a separate specimen).

When the image contains multiple clearly separated specimens as in Figure

4.6, it is trivial to separate the resulting masks and label each specimen independently. However, when multiple specimens overlap, separating them correctly becomes extremely difficult. Similarly, when a piece of the specimen is detached (e.g., a leaf or branch), it is difficult to determine whether this piece is part of the same specimen. Furthermore, many specimens are attached to the page with small pieces of tape. This tape is often not included in the foreground mask, resulting in disconnected specimens. Luckily, most of these issues were solved by dilating the masks first, then filtering, followed by an intersection with the original mask. The tape is sometimes (partly) included in the specimen mask, which is currently one of the main limitations of the semi-automatic labeling method.

After this process, the specimen masks were manually validated. Using an interactive script, we overlaid the masks on the original image and used keyboard inputs to quickly label the masks as correct or incorrect. The plant specimens were correctly extracted from 506 of the 1750 images (28.9%). The algorithm mainly failed due to the large variety of specimens, background colors, and objects in the dataset. For instance, some specimens had a larger surface area than the background, which incorrectly labeled their color as background. The most common failure occurred when parts of another object were contained within the specimen mask. This frequently occurred when they were positioned close to one another or overlapping.

We selected a sample of 250 images from the manually validated images and labeled seven common object classes (ruler, color card, note, barcode, stamp, attachment, and other) with their bounding polygons via the LabelMe annotation tool [24]. The class "note" details any attached note or textual information on the page. "Attachment" was used to label additional items such as envelopes, or attached fruit. The class "other" denotes rare objects, such as photographs. We noticed that many color cards contained a ruler, therefore this ruler strip was also labeled separately as a ruler. This way, there is no need for an additional class indicating the presence of both. Finally, these annotations were converted to the COCO [25] format and merged with the specimen masks, to construct a complete instance segmentation dataset. Figure 4.7 visualizes two fully labeled herbarium sheets from the training set.

From the 250 labeled images, 30 were randomly selected as part of the validation set used throughout this chapter. Table 4.2 lists the number of labeled objects for each class and the percentage of images on which they occur. Due to the low number of "other" objects (8 occurrences in 3 images), these were deliberately not included in the validation set. The other classes are relatively balanced, with "note" being the most prominent class. Surprisingly, some labeled images did not contain a ruler, while others had multiple.

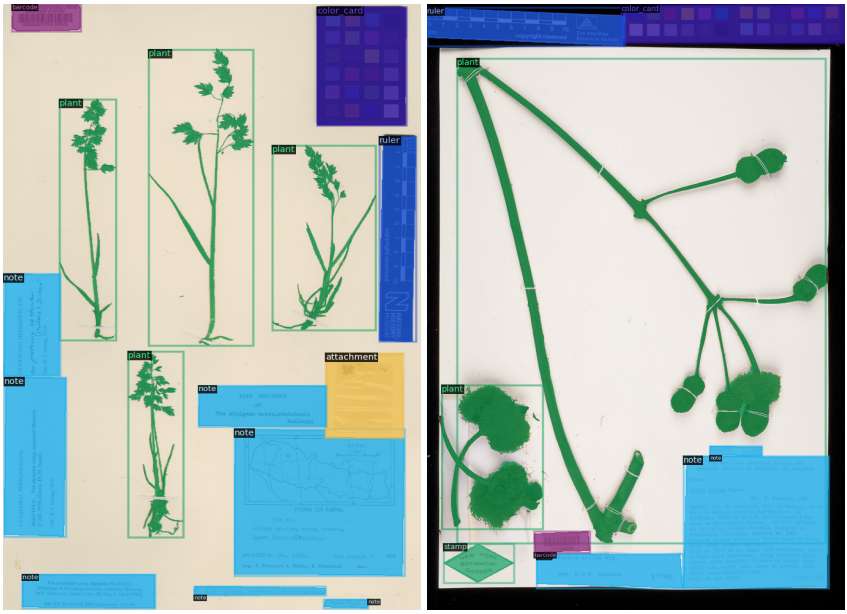


Figure 4.7: Two herbarium sheets from the training set with their labels visualized. Different instances of the same class are visualized with the same color to improve clarity. Best viewed in color and with zoom.

4.5.2 Binary Plant Segmentation

Three semantic segmentation models were trained to extract the plants from the background. These include U-Net, UNet++ [26] (an improved version of U-Net), and DeeplabV3+. Each model was trained with the same encoder, namely EfficientNet-B0 [27], starting from weights pretrained on Imagenet. The herbarium images and masks were resized to (608,800) and each model was trained for a maximum of 200 epochs. We used Dice loss to train the models and saved the model with the lowest validation loss. During training, the images were augmented with random color jittering, rotations, and reflections. Table 4.3 lists the intersection over union (IoU) and F_1 scores for each model on the validation set. The results show a large difference between the U-Net and DeeplabV3+ models. Even though the Unet++ architecture is more complex and was shown to outperform U-Net for biomedical image segmentation, we found little difference for the plant segmentation.

Table 4.2: Number of different labeled objects in the dataset and the percentage of images on which they occur.

Class	Train (%)	Validation (%)
Note	551 (99.5)	83 (100)
Barcode	231 (95.9)	34 (96.7)
Stamp	221 (80.5)	30 (80.0)
Ruler	216 (92.3)	29 (86.7)
Color card	145 (59.5)	21 (60.0)
Attachment	125 (56.8)	19 (63.3)
Other	8 (1.4)	0 (0)

Table 4.3: Binary plant segmentation results.

Model	IoU	F1
UNet++	**0.951**	**0.975**
U-Net	0.950	0.974
DeeplabV3+	0.915	0.954

4.5.3 Instance Segmentation

To detect both the plants and the other objects, we evaluated multiple instance segmentation models. These include YOLOv8l-seg (large), Mask R-CNN with FPN head, and Mask2Former (Swin-T backbone). For Mask R-CNN, we used the implementation from Detectron2 and also from Pytorch, both with a Resnet50 encoder. For YOLOv8 and Detectron2, their default training augmentations were used. For the Mask R-CNN and Mask2Former, we used custom data augmentations. Each model was trained for 200 epochs and the model with the lowest validation loss was saved.

To train the YOLOv8 model, the instance masks had to be converted to the YOLO format, namely a single polygon annotation per instance. This was performed using their supplied conversion script. However, many plant masks contain holes and have complex shapes, which makes the resulting polygon representations inaccurate.

The resulting average precision (AP) scores for both the bounding boxes and masks are presented in Table 4.4, as well as the mask AP for the plant class and the average mask AP of all other classes (object AP). These scores were calculated

Table 4.4: Results of the instance segmentation models.

Model	box AP	mask AP	mask AP50	plant AP	object AP
Detectron2	76.7	68.4	85.4	9.0	78.3
Mask R-CNN	78.2	76.7	92.7	31.9	**84.1**
YOLOv8	**87.0**	78.5	**96.1**	48.1	83.5
Mask2Former	80.7	**78.9**	91.0	**77.0**	79.2

using the official COCO evaluation code [7]. For the non-plant objects, all models performed relatively well, with mask AP scores ranging from 78.3 to 84.1. For the plant class, the Detectron2 and Mask R-CNN models scored poorly. Mask R-CNN scored slightly better, potentially due to the custom data augmentation. Their predictions often contained only a part of the entire plant. Conversely, the YOLOv8 model often predicted masks that were much bigger than the actual plant, which also led to poor performance. The Mask2Former model scored well on both plants and objects, making it a solid all-around choice. Figure 4.8 visualizes the predictions with a minimum confidence score of 0.5 for each model on a sample image from the validation set.

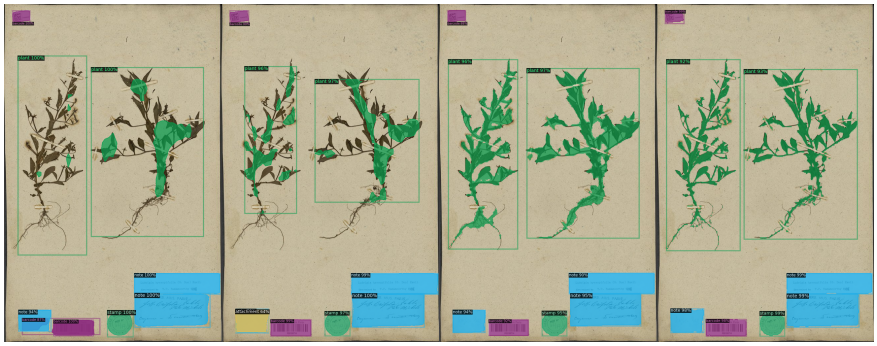


Figure 4.8: Predictions from each model on a sheet from the validation set. From left to right: Detectron2, Mask R-CNN, YOLOv8, and Mask2Former. Different instances of the same class are visualized with the same color to improve clarity. Best viewed in color and with zoom.

[7] https://cocodataset.org/#detection-eval

4.5.4 Panoptic Segmentation

Because most herbarium sheets only contain a single specimen, the problem can be reformulated as a panoptic segmentation task. The specimen(s) can be considered as a single "stuff" class (semantic segmentation) and the other objects as "things" (instance segmentation). So every pixel denoting a specimen will be labeled the same value, regardless of the number of specimens on the sheet. This way, predictions for the specimens will not be incorrectly split into multiple instances.

We have tested two approaches: a combined output of the previous UNet++ for the plant class paired with a retrained YOLOv8 for the objects and a single Mask2Former model, retrained on the panoptic labels. The same train and validation splits as before were used. Each model was trained for a maximum of 200 epochs. Table 4.5 shows the resulting mask AP scores for the objects and IoU scores for the plant class.

Table 4.5: Results of the panoptic segmentation approaches. Mask AP scores were calculated for the non-plant classes only.

Model	mask AP	mask AP50	plant IoU
YOLOv8 + UNet++	**83.7**	**98.3**	**0.951**
Mask2Former	81.6	95.7	0.899

The results show that the combined approach outperforms the Mask2Former model on both the objects and plant classes. Interestingly, the panoptic Mask2Former model performed slightly better on the objects than the previous instance segmentation model (mask AP of 81.6 vs 79.2). The combined approach achieved a mask AP of 83.7 and IoU for the plant class of 0.951, which are both better than the single Mask2Former model. Especially for the plant class, the difference in performance is clear. The Mask2Former model often struggled with segmenting smaller parts of the plants and objects. An example of this problem is shown in Figure 4.9, where predictions for both approaches are visualized (instance predictions were thresholded with a minimum score of 0.5).

4.6 Discussion

Our page and color card extraction pipeline has proven successful in assisting the digitization process. It greatly speeds up digitization and provides a standardized output that can be quickly manually verified. Detecting the color card via ORB

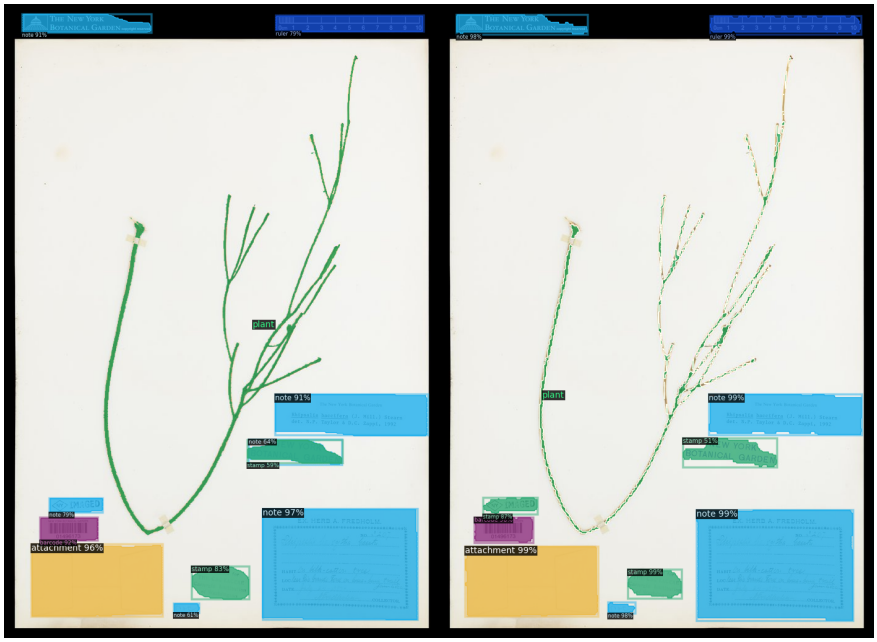


Figure 4.9: Panoptic predictions on a sheet from the validation set for the combined approach (YOLOv8 and Unet++) on the left and for Mask2Former on the right. Best viewed in color and with zoom.

features worked well, but was not consistent enough and made too many mistakes. By changing the digitization process slightly and enclosing the color card in a solid red color, we were able to achieve an accuracy of 97.6%. Some errors still occurred, which were either due to non-uniform sheets or human errors (e.g. color card not positioned correctly).

Automatic specimen identification was performed using exact and fuzzy string matching. Exact matching proved the most useful, achieving perfect results on around 40% of the images. Fuzzy matching lowered this precision, making it less applicable as a validation tool. We expect that more complex matching algorithms will achieve better results and can successfully include fuzzy matching. The matching speed was also not amazing, but popular tools like Elasticsearch[8] can be used to speed it up.

Our results for binary plant segmentation are in line with [13, 14, 18], which also achieved IoU and F_1 scores upwards of 0.95. We can conclude that the U-Net architecture can quickly and accurately segment plants from the background. However, instance segmentation proved a much more difficult task. Only the

[8] https://www.elastic.co/

Mask2Former model achieved a good segmentation of the plants, while the other models struggled. We suspect this is partly due to the Mask R-CNN architecture, which often smooths larger objects, removing the finer details [28]. The predictions from the YOLOv8 model were better, but still inaccurate, which was partly due to incorrect polygon labels.

For the non-plant objects, all models achieved a good performance, with mask APs ranging from 78.3 to 84.1. We suspect these results can be further improved by labeling additional data and using additional augmentation methods. Regarding processing time, the YOLOv8 model was the clear winner, which is an important consideration when processing large herbarium collections. The ruler class was generally the hardest to segment correctly, likely due to the many variations in the dataset. After post-processing, the extracted rulers can be used in combination with the plant masks to estimate the size and morphological traits of the specimens. Other objects can prove useful too, for instance, the notes can be cropped from multiple sheets and stacked into a single image. This reduces processing time for OCR tools and can often improve OCR results [29]. These results could then be used to improve or validate the manual data entry process during digitization.

By treating the segmentation as a panoptic segmentation task, we can leverage the performance and accuracy of the YOLOv8 and UNet++ models by combining their outputs. Such an approach achieves superior results compared to instance segmentation and is applicable when the herbaria sheets contain a single specimen or when semantic segmentation suffices for the plant class. We showed that this model combination outperformed a single panoptic Mask2Former model. Further tuning of the Mask2Former model can likely reduce this difference in accuracy. There are both benefits and drawbacks to using multiple models. The main drawback is that each model needs to be trained individually and then run separately for inference, which can increase processing time. Luckily, both UNet++ and YOLOv8 are quite performant and not memory-intensive. The benefit of using multiple models is that these can be trained on separate datasets. It is generally much easier to combine all the available binary plant segmentation datasets and retrain a plant segmentation model than it is to add object labels to these datasets, which are required to train a single panoptic model. This is also true for the instance segmentation model, although the used datasets would need to be normalized to contain the same object classes.

Because the binary segmentation models were trained on specimens labeled using the semi-automatic technique, this might bias the trained models and provide optimistic results. Therefore, we performed an additional qualitative evaluation of unlabeled specimens from the LifeCLEF dataset. We noticed that the segmentation results are generally comparable to the labeled dataset, but noticed some common segmentation errors. Plants were frequently not segmented

entirely or split into multiple parts. Other times the predicted masks were larger than the plant. Three examples of predictions on the unlabeled dataset containing such errors are given in Figure 4.10. These errors also occurred on the labeled dataset, but typically to a lesser extent.

Figure 4.10: Results of panoptic segmentation on the unlabeled dataset highlighting some common segmentation errors.

Regarding data labeling, the Segment Anything [30] model (SAM) or similar tools could be used to speed up the annotation of herbarium sheets. Our initial tests with SAM showed mixed results. Often, plants were split into multiple parts. This could however prove useful to annotate the specimens in a more detailed way, separating the leaves, branches, fruits, etc. [4] already showed promising results in calculating morphological features from leaves and we suspect such an approach can be generalized to additional parts of the specimens.

The main limiting factor for a more generic image processing approach is the lack of labeled image data [31]. It is a tedious and often difficult task to fully annotate herbarium sheets due to the diversity in species, objects, and quality. While this work introduced a novel instance segmentation dataset and promising results, additional research and labeled data are needed to further improve and evaluate the automated processing of herbarium sheets.

4.7 Conclusion

Our research has proven that automated herbaria processing tools can be developed using traditional computer vision methods and deep learning. The digitiza-

tion process can be sped up tremendously, by preprocessing or validating manually photographed specimens. OCR tools can be used to recognize plant names and match them with existing taxonomy databases. A semi-automatic labeling technique was developed and used to label a novel dataset of 250 digitized herbarium specimens with plant masks. Next, polygon annotations for 7 common herbarium objects were manually added. Different binary plant segmentation models were tested, with UNet++ achieving the highest IoU of 0.951. Four popular instance segmentation models were evaluated. YOLOv8l-seg and Mask R-CNN performed best on the object classes, achieving mask APs of 83.5 and 84.1, but they performed poorly on the plants. Mask2Former achieved the best overall results, with a mask AP of 78.9. The segmentation task was also reformulated as a panoptic segmentation problem, with the plant class as a semantic class. A combination of YOLOv8 and UNet++ outperformed the Mask2Former model, achieving a higher IoU for the plant class and a higher mask AP for the non-plant objects. While these results are promising, further research and labeled data are needed to improve and evaluate the automated processing of herbarium specimens on a larger scale.

References

[1] B. M. Thiers. *The World's Herbaria 2021: A Summary Report Based on Data from Index Herbariorum*. https://sweetgum.nybg.org/science/wp-content/uploads/2022/02/The_Worlds_Herbaria_Jan_2022.pdf, 2022.

[2] B. M. Thiers, M. C. Tulig, and K. A. Watson. *Digitization of the new york botanical garden herbarium*. Brittonia, 68:324–333, 2016.

[3] J. Gaikwad, A. Triki, and B. Bouaziz. *Measuring Morphological Functional Leaf Traits From Digitized Herbarium Specimens Using TraitEx Software*. Biodiversity Information Science and Standards, 3:e37091, 2019. arXiv:https://doi.org/10.3897/biss.3.37091, doi:10.3897/biss.3.37091.

[4] A. Triki, B. Bouaziz, J. Gaikwad, and W. Mahdi. *Deep leaf: Mask R-CNN based leaf detection and segmentation from digitized herbarium specimen images*. Pattern Recognition Letters, 150:76–83, 2021.

[5] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241, Cham, 2015. Springer International Publishing.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, Computer Vision – ECCV 2018, pages 833–851, Cham, 2018. Springer International Publishing.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. *You only look once: Unified, real-time object detection*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[8] G. Jocher, A. Chaurasia, and J. Qiu. *YOLO by Ultralytics*, January 2023. Available from: https://github.com/ultralytics/ultralytics.

[9] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2, 2019.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. *Mask r-cnn*. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[11] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. *Masked-Attention Mask Transformer for Universal Image Segmentation*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1290–1299, June 2022.

[12] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi. *Oneformer: One transformer to rule universal image segmentation*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2989–2998, 2023.

[13] A. E. White, R. B. Dikow, M. Baugh, A. Jenkins, and P. B. Frandsen. *Generating segmentation masks of herbarium specimens and a data set for training segmentation models using deep learning*. Applications in Plant Sciences, 8(6):e11352, 2020.

[14] B. R. Hussein, O. A. Malik, W.-H. Ong, and J. W. F. Slik. *Semantic Segmentation of Herbarium Specimens Using Deep Learning Techniques*. In R. Alfred, Y. Lim, H. Haviluddin, and C. K. On, editors, Computational Science and Technology, pages 321–330, Singapore, 2020. Springer Singapore.

[15] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. *Full-resolution residual networks for semantic segmentation in street scenes*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4151–4160, 2017.

[16] R. J. Wilson, A. F. de Siqueira, S. J. Brooks, B. W. Price, L. M. Simon, S. J. van der Walt, and P. B. Fenberg. *Applying computer vision to digitised natural history collections for climate change research: Temperature-size responses in British butterflies*. Methods in Ecology and Evolution, 14(2):372–384, 2023.

[17] Abdelaziz, Bassem, and Walid. *A deep learning-based approach for detecting plant organs from digitized herbarium specimen images*. Ecological Informatics, 69:101590, 2022. Available from: https://www.sciencedirect.com/science/article/pii/S1574954122000395, doi:10.1016/j.ecoinf.2022.101590.

[18] K. Milleville, K. K. Thirukokaranam Chandrasekar, and S. Verstockt. *Automatic Extraction of Specimens from Multi-specimen Herbaria*. ACM Journal on Computing and Cultural Heritage, 16(1):1–15, 2023.

[19] Thirukokaranam Chandrasekar, Krishna Kumar. *Meta data enrichment for improving the quality and usability of botanical collections*. PhD thesis, Ghent University, 2022.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. *ORB: An efficient alternative to SIFT or SURF*. In 2011 International Conference on Computer Vision, pages 2564–2571, 2011. doi:10.1109/ICCV.2011.6126544.

[21] H. Goëau, P. Bonnet, and A. Joly. *Overview of LifeCLEF Plant Identification task 2020*. In CLEF 2020-Conference and Labs of the Evaluation Forum, volume 2696, 2020.

[22] G. Secretariat. *GBIF Backbone Taxonomy*, 2021. Available from: https://doi.org/10.15468/39omei.

[23] M. Dillen, Q. Groom, S. Chagnoux, A. Güntsch, A. Hardisty, E. Haston, L. Livermore, V. Runnel, L. Schulman, L. Willemse, et al. *A benchmark dataset of herbarium specimen images with label data*. Biodiversity Data Journal, (7), 2019.

[24] K. Wada. *Labelme: Image Polygonal Annotation with Python*. Available from: https://github.com/wkentaro/labelme.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. *Microsoft COCO: Common Objects in Context*. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740–755, Cham, 2014. Springer International Publishing.

[26] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. In D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, editors, Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pages 3–11, Cham, 2018. Springer International Publishing.

[27] M. Tan and Q. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. In K. Chaudhuri and R. Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. Available from: https://proceedings.mlr.press/v97/tan19a.html.

[28] A. Kirillov, Y. Wu, K. He, and R. Girshick. *Pointrend: Image segmentation as rendering*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9799–9808, 2020.

[29] D. Owen, L. Livermore, Q. Groom, A. Hardisty, T. Leegwater, M. van Walsum, N. Wijkamp, and I. Spasić. *Towards a scientific workflow featuring Natural Language Processing for the digitisation of natural history collections*. Research Ideas and Outcomes, 6:e55789, 2020. arXiv:https://doi.org/10.3897/rio.6.e55789, doi:10.3897/rio.6.e55789.

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. *Segment anything*. arXiv preprint arXiv:2304.02643, 2023. Available from: https://arxiv.org/abs/2304.02643.

[31] K. D. Pearson, G. Nelson, M. F. J. Aronson, P. Bonnet, L. Brenskelle, C. C. Davis, E. G. Denny, E. R. Ellwood, H. Goëau, J. M. Heberling, A. Joly, T. Lorieul, S. J. Mazer, E. K. Meineke, B. J. Stucky, P. Sweeney, A. E. White, and P. S. Soltis. *Machine Learning Using Digitized Herbarium Specimens to Advance Phenological Research*. BioScience, 70(7):610–620, 05 2020. arXiv:https://academic.oup.com/bioscience/article-pdf/70/7/610/33476139/biaa044.pdf, doi:10.1093/biosci/biaa044.

5

Processing Textual data

"The purpose of computing is insight, not numbers"

– Richard Hamming

This chapter discusses the automated processing of textual data using natural language processing methods. Such methods enable the extraction of semantic information from text. We applied these methods to Twitter data to analyze spatiotemporal differences in the public opinion surrounding forest fires and renewable energy. The chapter concludes with a case study, where Flickr data was used to analyze tourism interests in the cities of Ghent and Vienna.

This chapter features an adapted version of the following publications:

Milleville, K., Van Ackere, S., Verdoodt, J., Verstockt, S., De Maeyer, P., & Van de Weghe, N. (2023). **Exploring the potential of social media to study environmental topics and natural disasters.** JOURNAL OF LOCATION BASED SERVICES. https://doi.org/10.1080/17489725.2023.2238663

Milleville, K., Ali, D., Porras-Bernardez, F., Verstockt, S., Van de Weghe, N., & Gartner, G. (2019). **WordCrowd: a location-based application to explore the city based on geo-social media and semantics.** In G. Gartner & H. Huang (Eds.), Adjunct proceedings of the 15th international conference on location based services (LBS 2019) (pp. 231–236). https://doi.org/10.34726/LBS2019.29

5.1 Introduction

Textual data can be found in many digital collections or it can be the result of processing such collections (e.g. OCR on digitized newspapers). Often, this type of data is unstructured and used to search through the collection. This can make it difficult to efficiently query the collection and often too many results are returned. Natural language processing techniques can be used to analyze this data automatically. Such techniques can be used to geolocate place names, determine the topic of the text, extract named entities, summarize the text, and much more. The techniques are generic and can typically be applied to any textual data source.

In our research, we have successfully applied many different NLP techniques to social media posts, OCR results of digital archives, image descriptions, and more. Many of these collections contained geospatial data, which provided additional dimensions to the data. This geospatial data was either provided as additional metadata in a structured format (timestamp or coordinate information) or was embedded in the text itself (place names or dates). After processing, the results can be analyzed and visualized spatially, temporally, or semantically. By leveraging these spatiotemporal dimensions, we can uncover trends and detect outliers in the collections.

5.2 Geospatial Data

In recent years, governments at regional, national, and supranational levels have invested heavily in the standardization (e.g., Inspire), collection (e.g., Copernicus), and use of remote sensing data to monitor environmental parameters. The qualitative development of geospatial information technologies and services over the past two decades has led to a dramatic increase in the amount of data that can be used to assess the state of the environment. Although the amount of data has increased substantially, the quality of decisions made based on this data has not improved much [1]. Remote sensing data collection is typically performed using satellite imagery. This method requires significant investment in hardware and software to process the data into indicators. Various characteristics describing the ecological condition of areas have been derived from satellite images taken at different moments in time, resulting in spatiotemporal indicators [2]. Atlas information systems and their qualitative advancements, such as Digital Earth, are becoming increasingly important, especially in the field of environmental management. New sources of information and new approaches to aggregate and analyze information greatly expand the ability to monitor the ecological condition of areas and support environmental decision-making.

In addition to remote sensing data, new types of data are being incorporated

into environmental studies. Research in citizen science, crowdsourcing, and social media has shown the potential to gather both knowledge and insights to address environmental emergencies [3, 4]. Although the use of social media data is useful for disaster management, further research is needed to sift through the (potentially) interesting information and provide valuable insights for environmental indicators [5].

The biggest challenges lie in collecting and aggregating the myriad of data across different technologies. As each social media post in itself provides little additional information, they must be aggregated both semantically (via natural language processing) and spatiotemporally (via clustering techniques). Once the data is aggregated, the next step is to explore the transformation of this information into valuable indicators at local and global levels.

Using social media data, we aim to provide an additional dimension to environmental data by analyzing public opinion about different environmental topics and studying the impact of natural disasters. We expect to find regional differences on these topics and want to examine how public opinion has changed over time. Do people think environmental topics, such as global warming and renewable energy, are important? If so, when did this change occur, and can we hypothesize about what caused this change in mentality? Concerning natural disasters, we will study their immediate impact online and determine whether they had a long-term impact on people's views. For instance, did the forest fires around the world reignite the discussion about global warming? Did the tragedy in Fukushima lead to a negative change in mentality about nuclear energy? Especially when it comes to polarizing topics, this data can provide new insights. We hope to answer most of these questions and provide a framework that will allow other researchers and legislators to freely access the data and gain insights to answer similar questions.

A similar approach was used in a case study on metadata from pictures posted on Flickr in the cities of Ghent and Vienna. Using this metadata, we were able to determine tourism hotspots and spatiotemporal differences in tourism interests. When grouping the tourists by their country of origin, we uncovered different hotspots and interests automatically. The results of the case study were visualized in an interactive LBS (Location-Based Service) application.

5.2.1 Related Work

To automatically analyze large text datasets, natural language processing techniques can be used. NLP is the field of computer science that deals with the automatic extraction of information from (unstructured) text [6]. With recent advances in neural network architectures, faster computing, and larger datasets to train on, the field has made tremendous progress. Many NLP models can be used

immediately on new, unseen datasets and still perform well. Sentiment analysis is a popular NLP technique to predict an author's sentiment from their text. Usually, sentiment is denoted as a class (positive, negative, or neutral) or as a numerical value (-1 to 1). For most people, it is often straightforward to determine the sentiment of a given text. However, due to the complexity of natural language, the small amount of text per tweet, sarcasm, and the unique vocabulary used in certain subcultures, it can be difficult to determine sentiment in an automated way.

A related problem is stance detection, which involves determining whether the author is in favor, against, or neutral toward a given statement or topic. The author's stance can either be explicitly mentioned or implied in the text. In [7], this problem was posed as a supervised learning task by annotating a dataset of tweets on five different topics: atheism, climate change, feminism, Hillary Clinton, and abortion. The annotated data was used to create the SemEval-2016 Task 6 challenge. The highest F_{avg} score of the participating teams was 67.8, indicating that this problem is not easy to solve. In recent work, these scores have been improved by using large pre-trained language models. One popular model, BERT (Bidirectional Encoder Representations from Transformers), is a state-of-the-art language model that uses a bidirectional transformer architecture [8]. This model was pre-trained on huge corpora of unlabeled text, allowing rapid fine-tuning on a wide range of NLP tasks. Even though these models can outperform traditional techniques, they typically require labeled data for each specific topic or statement to fine-tune. Automatically determining the specific topic or statement being talked about positively or negatively is even more challenging.

In the event of a (natural) disaster, social media users produce many posts with disaster-related information that can be useful for analysis [9–11]. For instance, [12] found that extracting sentiments during Hurricane Sandy could help emergency responders develop better situational awareness of the disaster area. In another study, a BERT model was applied to a set of tweets related to the Jakarta floods in early 2020 to identify relevant tweets that could provide information on disaster response [13]. Besides natural disasters, previous research has shown correlations between mode of travel and tweet sentiment [14], between temperature anomalies and tweeting behavior [15], and between people's concern about climate change and the severity of weather anomalies [16]. By analyzing tweets made with the hashtag #WorldEnvironmentDay, [17] found that certain environmental topics (climate change, clean water, and pollution) carried a negative sentiment. While other topics such as public health and clean energy, were rather positive. These results can potentially be used by NGOs or policymakers to focus on the most concerning topics. In [18], a semi-automatic method was developed to label over 20,000 tweets regarding their stance on the independence of Catalonia. Their method greatly speeded up the labeling process by

leveraging user-based relations. Their best models were based on the BERT architecture and achieved an F_{avg} score of 0.7468 and 0.7472 on Catalan and Spanish tweets, respectively.

More recently, large language models such as ChatGPT or GPT-4, are being used for sentiment analysis or stance detection. These models can be applied in a zero-shot fashion (using no labeled data or examples) and achieve close to state-of-the-art performance on certain benchmarks [19, 20]. One study even found that ChatGPT outperformed crowd workers to label tweets, for a fraction of the cost [21]. While these large language models are computationally intensive, they are becoming increasingly popular to tackle a wide range of NLP tasks.

5.2.2 Twitter Data

The proposed approach for investigating public opinion and spatiotemporal differences related to natural disasters consists of four phases: tweet collection, tweet processing, tweet georeferencing, and analysis.

5.2.3 Collection and Preprocessing

Over the past decade, social media has become an integral part of global communication and is therefore frequently used as a data source for large-scale analyses. Twitter, a social network where users can send tweets (short messages of up to 140 characters, increased to 280 characters in 2017), is often used to collect such data. It has an open API for research purposes with detailed query functionalities. It is estimated that over 500 million tweets are sent daily, allowing for extensive data collection on virtually any topic. However, many people do not use Twitter, preferring alternatives such as Facebook, Reddit, and Sina Weibo (popular in China). We are aware that these alternatives exist and that their exclusion could lead to geospatial bias, but we will focus only on Twitter data to limit the scope of the project. However, the proposed methods can be applied to virtually any social media platform.

First, a query relating to forest fires was performed in four languages: English, Spanish, Chinese, and Russian. Each query consisted of multiple writing variations on the topic of forest fires. The tweets were collected from January 2012 until August 2021. The last two years of tweets were processed and analyzed in detail. Table 5.1 shows the number of retrieved tweets in those last two years, the percentage of geotagged tweets, the number of user locations found, and the percentage that was successfully geolocated using our algorithm (see 5.2.3.1). Most tweets are written in English or contain some English keywords. Although Chinese is the second most popular language in the world, we found almost 30

times more tweets searching with English keywords. This is likely because Twitter is less popular in Chinese-speaking countries and due to translation errors. Additionally, many viral keywords, hashtags, and trends are often written in English. This causes many non-native English speakers to tweet in English or use such keywords to increase their reach. Using the estimated language provided by Twitter, we find that 94% of the tweets found with English keywords were written in English.

Table 5.1: The number of tweets and user locations found for each query language

Language	Total tweets (thousands)	Geotagged tweets (%)	Locations (thousands)	Geolocated locations (%)
English	2,465	2.32	1,890 (76.7%)	69.7
Spanish	433	2.62	349 (80.6%)	73.8
Chinese	84	0.45	46 (54.4%)	12.4
Russian	17	1.03	12 (69.0%)	28.0

Because tweets can be sent in any language, this complicates both the retrieval of tweets and their analysis. A popular approach is to translate all collected tweets into a common language (usually English) and then process them using language-specific models. To query topics in different languages, keywords or hashtags have to be translated using available translation services and models. However, sometimes these automatic translations do not reflect the correct translation, or there are several common spellings for the specified topic. For instance, if you aim to collect tweets regarding the coronavirus epidemic, you should use multiple related keywords, such as "covid", "coronavirus", and "corona epidemic". Even after searching for all possible spelling variations of the topic, many tweets related to the topic may not be found. The collected tweets consisted of multiple fields that contained additional information about the tweet itself. Table 5.2 lists some of the most important fields with their explanations.

To perform a large-scale spatiotemporal analysis, we require coordinate information. Twitter allows users to tag their tweets with the exact coordinates of their location (geotag). Our findings show that less than 2.5% of all collected tweets were geotagged. Therefore, we need to rely on the location provided in the user's Twitter profile to determine an approximate location. Between 60-80% of users provided their location in plain text. Users can enter anything as their location, so many locations will not provide useful information. For instance, one query found that over 10% of users listed their location as “earth” or “planet earth”. To determine the coordinates that relate to these location names, geocoders can be used.

Table 5.2: Description of the tweet fields used in this work.

Field name	Explanation
text	The content of the tweet
lang	Language of the tweet, detected by Twitter
created_at	Creation time of the tweet
id	Unique identifier of this tweet
author_id	Unique identifier of this user
user_location	(Optional) User-submitted profile location in plain text
geo	(Optional) Details & coordinates of the geotagged location

5.2.3.1 Geocoding

Geocoding is the process of transforming a location description (address or place name) into a coordinate on the earth's surface. There are a variety of algorithms for geocoding, but they all follow roughly the same process. First, the address to be geocoded is entered in plain text. Then, the address is normalized into an acceptable format (usually street name, house number, city name, and postal code). Finally, an iterative comparison of this address with a reference dataset (e.g., a street and city database) is performed, from which the geographic coordinates of the address can be calculated [22].

Geocoders are usually accessed via a REST API. Popular examples include Google Geocoding[1] and the open-source solutions Geonames[2] and Nominatim[3]. These APIs have tight limitations and can become expensive to geocode millions of user locations. Therefore, we developed a simple algorithm to geocode popular locations and place names. Our method used a reference dataset containing all countries, their main cities, and provinces[4]. In total, this dataset contained about 43,000 place names. An iterative algorithm was developed that considered exact string matches of the found place names with this reference dataset.

First, the user location was queried for a country name. If it contained a country name, we checked if it contained the name of a city or province/state of that country. If it did, the coordinates were extracted, favoring cities over provinces, as they provide more localized information. If no country was found, we checked for a matching province or state and city name (e.g., "Nashville, TN" was matched to "Nashville, Texas"). If no province or state was found, we queried for just a

[1] https://developers.google.com/maps/documentation/geocoding/overview

[2] https://www.geonames.org/

[3] https://nominatim.org/

[4] https://simplemaps.com/data/world-cities

city name. If multiple matches were found with the same city name, the most populous option was chosen (e.g., "Paris" was matched to "Paris, France" rather than "Paris, Texas"). This disambiguation could be further improved by considering the language of the tweet and the native language of the matching countries. The algorithm worked quite well with place names written in the Latin alphabet and was able to geocode 70.3% of the user locations (see Table 5.1). However, it performed poorly in the initial tests with other alphabets (Russian and Chinese). This is because the reference dataset often does not contain place names in the local alphabet. After geocoding all unique user locations with this algorithm, the locations that did not result in a match can be geocoded with a public API, greatly reducing the number of requests.

5.2.3.2 Sentiment Analysis and Stance Detection

After processing and geocoding the collected tweets, sentiment analysis was performed using Textblob. Textblob is a popular sentiment analysis model, available as an open-source Python library [23]. In addition to sentiment analysis, the library provides a consistent API for common NLP tasks such as part-of-speech tagging, noun phrase extraction, and more. Textblob determines the sentiment with a predefined dictionary that classifies negative and positive words. All words in the analyzed sentence receive an individual score depending on whether they are positive or negative. A pooling operation, such as the average of all sentiments, is then used to calculate the final sentiment. TextBlob provides two types of information about the input sentiment: polarity and subjectivity. Polarity ranges from [-1,1], where -1 represents a negative sentiment and 1 represents a positive sentiment. Subjectivity ranges from [0,1] and tries to distinguish facts from opinions. Higher subjectivity means that the text contains personal opinions rather than factual information. The tweets were first preprocessed by removing all mentions (@username), URLs, and hashtag symbols. Then, both polarity and subjectivity were calculated using Textblob. Using the predicted polarity, the goal was to visualize how public opinion varies spatiotemporally. We found that tweets related to natural disasters were not very polarizing and were difficult to analyze after aggregation on a large scale (see Section 5.2.4).

Therefore, an additional test was conducted on tweets about alternative energy sources (nuclear, solar, wind), which represented a more polarizing topic. Instead of using a generic sentiment analysis model, we fine-tuned a language model for stance detection on a small subset of the collected tweets related to nuclear energy. The tweets were manually labeled as either in favor, against, or neutral (neither) towards nuclear energy as an alternative energy source. During the labeling process, we found many irrelevant tweets. Some discussed nuclear weapons, some were job ads, and some had nothing to do with nuclear energy but

contained one or more keywords. Due to the large number of irrelevant tweets, we added irrelevance as an additional label. A total of 500 tweets were labeled using the open-source tool Label Studio [24]. These labeled tweets were then used to fine-tune a BERTweet model [25], which is a pre-trained language model that uses a similar architecture as BERT. BERTweet was pre-trained on large corpora of English tweets and outperformed other pre-trained models on NLP tasks on tweets. In addition, the model and code are released under an open-source license.

5.2.4 Results

5.2.4.1 Sentiment Analysis on Forest Fires

To compare our results related to wildfires, the international disaster database EM-DAT[5] of the Centre for Research on the Epidemiology of Disasters was used. All Spanish tweets related to wildfires (from January 2012 to July 2021) were collected, geocoded, and filtered for tweets posted from Spain. Figure 5.1 shows these tweets along with some of the major wildfires in Spain reported in the EM-DAT database. There is clearly a recurring pattern in posts about forest fires during the summer. There is a clear overlap between the Twitter data and the wildfire occurrences, whether at the local (in Spain) or global level. For instance, four peaks in the Twitter data correspond to reported wildfires in Spain: July 2012, June 2017, October 2017, and July 2021.

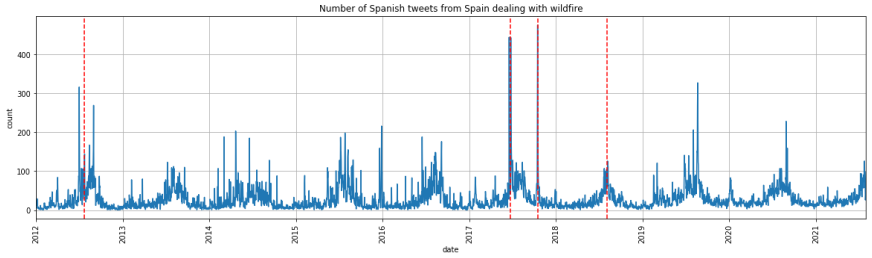


Figure 5.1: Number of Spanish tweets from Spain dealing related to wildfires, red dashed lines represent some major reported wildfires in Spain from EM-DAT

Remarkably, the large spike in Spanish tweets in August 2019 did not coincide with a reported wildfire in Spain. The only major wildfires reported in the EM-DAT database were in Australia (New South Wales, Queensland). Looking at the English tweets posted in Europe dealing with wildfires, this peak is also noticeable. We

[5] http://www.emdat.be/database

can conclude that it is possible to detect the occurrence of wildfires using Twitter data. However, user locations do not indicate where these wildfires are occurring, as many people across the globe tweet about major wildfires. Further analysis of the text content is needed to determine the wildfire location.

Figure 5.2 shows the complementary spatial distributions of Spanish and English tweets related to wildfires. There is a clear correlation between Spanish user locations and countries where Spanish is the official native language. For the English query, we see better global coverage, showing that these English keywords and hashtags are used by many non-native speakers. Furthermore, we found that over 99% of the tweets found with the Spanish query were unique and not included in the English query.

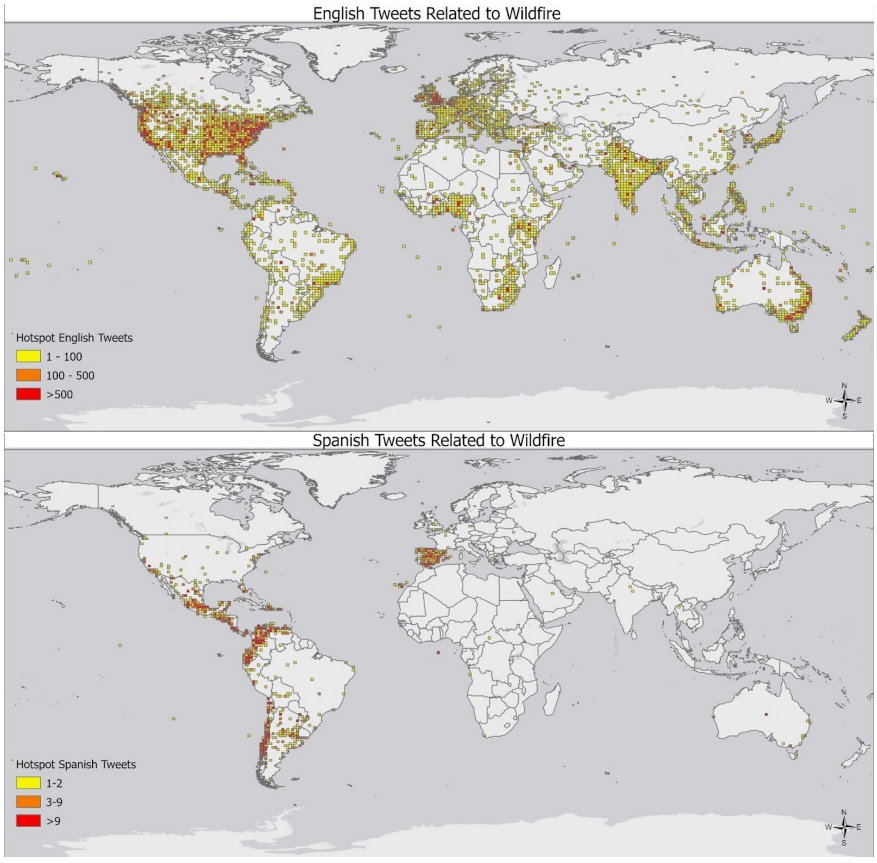


Figure 5.2: The distribution of English (top) and Spanish (bottom) tweets related to wildfires

After taking a closer look at many tweets and their predicted sentiment, we

concluded that the Textblob model cannot accurately assess sentiment. Many tweets contain relevant keywords or hashtags (e.g., forest fire, wildfire), but are irrelevant to the topic. We suspect that many of the viral hashtags related to wildfires are used to gain more reach for an individual's tweets, even though the tweet is unrelated to wildfires. These irrelevant tweets heavily influence the results, therefore, they either need to be filtered out beforehand or the model should ignore them. Some positive and negative predicted tweets are shown in Table 5.3. Interestingly, the model predicted many tweets with a positive sentiment, most of which were irrelevant concerning wildfires. Furthermore, when the sentiments are aggregated over large areas, they tend to average out to neutral and provide little insight into public opinion.

5.2.4.2 Stance Detection on Nuclear Energy

Out of the 500 manually labeled tweets for stance detection regarding nuclear energy, 98 were labeled as irrelevant. Of the 402 others, 169 were labeled as "in favor", 86 as "neither", and 147 as "against". Two tests were performed: one to predict the tweet's relevance and one to predict the author's stance with respect to nuclear energy. For both tests, 20% of the data was used for validation (100 tweets). The relevance prediction performed surprisingly well, with an F_1 score of 0.92. The fine-tuned model was clearly able to distinguish tweets related to nuclear energy from unrelated tweets.

Because we are mainly interested in favorable or negative opinions, the irrelevant tweets were considered as "neither" for stance detection. To evaluate the overall performance, we used the macro-average of the F_1 scores (denoted as F_{avg}) for the "in favor" and "against" classes. This is the same metric that was used in [7]. The stance detection was less accurate than the relevance prediction with an F_{avg} of 0.67. The model was also much better at predicting the favorable class. For completeness, Table 5.4 lists the precision, recall, and F_1 scores for each class in the validation set.

Taking a closer look at some of the incorrect predictions on the validation set, we saw that the model sometimes made confident mistakes. Other times, none of the predictions had a high probability, so these could be ignored by using a threshold. For instance, if we only consider predictions with a minimum threshold of 0.75, the F_{avg} score rises to 0.765, but at the cost of discarding 52 % of the tweets in the validation set. Some example tweets with incorrect predictions are presented in table 5.5.

Table 5.3: Some sample tweets related to wildfires, grouped by their predicted sentiment. Many of the collected tweets contained viral hashtags related to wildfires but were irrelevant.

Tweets with negative predicted sentiment	Irrelevant
Sam Wood and Snezana Markoski raise $20,000 in donations for bushfire relief in just 24 hours The Bachelor's Sam Wood and Snezana Markoski are doing their part in helping Australians affected by the devastating bushfire crisis	
There will be a day That all the diabolical and evil deeds of these politicians will be met by a raging wildfire that will engulf them and riches they have robbed this nation of.	X
Neguse Curtis Launch Bipartisan Wildfire Caucus Introduce Legislation to Help Communities Recover From Devastating 2020 Wildfire Season. TY Sen Neguse!	
My brain cannot wrap itself around a fire crossing the continental divide How can a wildfire reach 11-12,000 feet Absolutely insane.	

Tweets with positive predicted sentiment	Irrelevant
You can find them best the year after a forest fire.	X
Best of luck to all the nominees #Wolfwalkers #DatingAmber #Wildfire #Vivarium #HereAreTheYoungMen #SeaFever	X
Beautiful sunrise underway in Missoula courtesy of the wildfire smoke! You can expect hazy skies again today but lessening going into tomorrow MTwx	
Man this bird is awesome #lyrebirds #leonardthelyrebird #bluemountains #AustralianBushfires	X
What a brilliant idea watch it catch on like wildfire!	X

5.2.5 Discussion

In this Section, we presented a generic pipeline for spatiotemporal analysis of tweets on environmental topics and our preliminary results. We showed that simple sentiment analysis models often underperform on tweets. Furthermore, the predicted sentiment does not provide sufficient information to perform an in-depth analysis of public opinion when aggregated over larger regions.

The relatively simple geocoding algorithm was able to geocode 70.3% of the collected tweets in the Latin alphabet by using the locations of the users in their

Table 5.4: Validation scores for each class of the stance detection

Stance	Precision	Recall	F_1 score
Against	0.63	0.63	0.63
Neither	0.81	0.69	0.75
In favor	0.67	0.76	0.71

Twitter bio. The locations that did not yield a match can be geocoded using a public API, greatly reducing the number of queries. These user locations are critical because less than 2.5% of all tweets were geotagged. However, when using geocoding APIs, certain user locations such as "earth" and "nowhere" can match a real place name, resulting in false positives. Automatically removing these false positive matches will be a challenge.

Upon closer examination of the collected tweets, we found that many of them were irrelevant to the queried topic. Many news reports, job ads, or tweets on similar topics (e.g., nuclear weapons) contained some of the keywords. The inclusion of these irrelevant tweets will lead to an overestimation of the number of tweets and people discussing the topics at hand. However, we showed that it is possible to accurately filter out irrelevant tweets by fine-tuning a language model. While this approach produced good results, it can be time-consuming when applied to the full dataset of millions of tweets. Additionally, this approach was tested for a single topic (nuclear energy). Future research will show whether a single model can be used to filter out most irrelevant tweets across topics, or whether a separate model is needed for each topic. We estimate that news reports, job ads, financial information, and other similarly structured irrelevant tweets can be automatically filtered out.

The retrained BERTweet model for stance detection regarding nuclear energy achieved an F_{avg} score of 0.67. Considering that the model was only trained on 400 tweets and validated on the remaining 100, this is a promising result. When analyzing the incorrect model predictions, we saw that many of them were replies to another tweet, were too short, or were written in a convoluted way where it is difficult to determine the stance without additional context. These problems were also mentioned in [18, 26]. Although our analysis focused solely on tweets, the discussed methods can be applied with little adjustments to other social media platforms featuring text-based content such as Facebook and Reddit.

Our goal is to label additional data for stance detection in queries about alternative energy sources (nuclear, solar, wind, etc.) to visualize the spatiotemporal evolution of public opinion over the last decade. We will investigate the use of large language models like ChatGPT and GPT-4 to speed up the labeling process,

Table 5.5: Sample tweets from the validation set with incorrect predictions and associated scores and labels.

Tweet text	Prediction	Label
I've been reading a book about the Chernobyl accident and it's had me thinking. Considering how the Russian government botched the building and managing of those reactors, imaging the disaster if the trump admin were to attempt something like nuclear energy.	Neither (0.848)	Against
@CKscullycat Not to mention, nuclear power plants	Against (0.696)	Neither
Observing the #printergate debacle, I think it was wise we eschewed nuclear energy.	Favor (0.449)	Against
@GavinNewsom How about spending money on infrastructure, nuclear power, etc... to accommodate the CA population's need for energy? Just like H2O, with proper planning these "emergencies" can be avoided	Against (0.575)	Favor
went down a nuclear energy rabbit hole tonight like how did we not ditch the whole "atomic age" thing after chernobyl? fukushima?? we're really still out here burying radioactive waste in concrete sarcophaguses in 2020? wild	Neither (0.393)	Against

as these offer exceptional zero and few-shot performance [19–21]. The resulting dataset will be anonymized and published with a permissive license to stimulate further research. We also intend to conduct a small study of the model's performance on tweets that were automatically translated into English. This translation is likely to affect the performance of the model, which is important if we are to include multilingual queries.

5.3 Case study: Wordcrowd

WordCrowd[6] is a dynamic location-based service that visualizes and analyzes geolocated social media data. By spatially clustering the data, areas of interest and their descriptions can be extracted and compared on different geographical scales. When walking through the city, the application visualizes the nearest areas of interest and presents these in a word cloud. By aggregating the data based on the country of origin of the original poster, we discover differences and similarities in tourist interests between different countries. This case study was part of Eureca, a collaborative project with the cartography research group from the Technical University of Vienna (TU Wien) and several city and state archives from Ghent and Vienna.

A post on social media reflects the thoughts and feelings of the poster about a certain topic as a data point in space and time. By focusing on the location of the post instead of on the content, areas of interest (AOIs) can be extracted as areas with a higher post density. By spatially clustering these points, these AOIs are automatically extracted and most of the noise is filtered out. The dataset used for this research consists of geolocated Flickr pictures and their associated tags. It covers continental Europe with metadata of all the images uploaded from 2004 to 2018. Nevertheless, our approach and application can work with any type of geolocated textual data.

The clustering technique is an essential part of this analysis and modifying it will impact the number of AOIs, their size, shape, and contents. In this research, HDBSCAN [27] is used as the main clustering algorithm. HDBSCAN is an extension of the popular DBSCAN algorithm and performs better on datasets with varying densities. By changing the parameters of the algorithm, the clustering can be performed on different scales. This multi-scale clustering is necessary for an interactive LBS application, as the user might be overwhelmed with a large number of smaller clusters when he zooms out on the map. To ensure the application works in real-time with a dynamic interface, we have preprocessed the data into multi-scale clusters and visualized only the nearby clusters instead of all the nearby points.

When the user zooms out, the application will fetch and visualize the larger nearby clusters from the database to reduce the network and memory load. Figure 5.3 shows this functionality. Clicking an AOI displays its aggregated tags in a word cloud. This provides an intuitive visualization of the tags contained in each AOI. Initially, the points located in Austria (370,000 points) were clustered on three scales, resulting in 845, 79, and 8 clusters. For each cluster, we aggregate and preprocess all the related tags. The top 100 most frequently occurring tags and

[6] https://labeltool.idlab.ugent.be/wordcrowd/map/

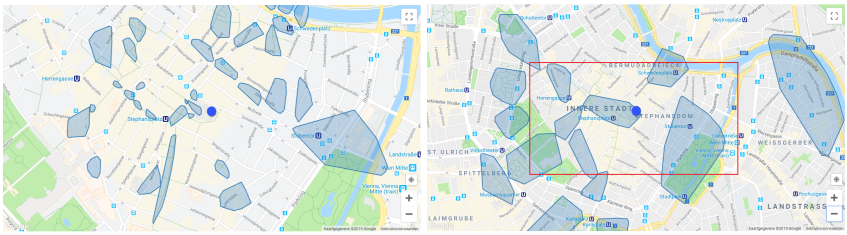


Figure 5.3: Visualization of the smallest clusters for a part of Vienna (left) and the larger clusters when the user zooms out (right). The user's current position is marked with a blue dot.

their frequency are then saved in our database. This gives us geolocated AOIs and their descriptions generated from the Flickr picture tags. Afterward, this process was repeated for all the points located in Belgium (430,000 points).

Because we were dealing with very noisy and multilingual picture tags, these needed to be preprocessed to improve the generated word clouds. First, all the tags were translated into English to provide a common language for the following preprocessing steps. Next, irrelevant tags like brand names and stop words were removed. Afterward, traditional NLP techniques such as sequence matching, stemming, and lemmatization were used to group similar words together. Finally, redundant multilingual place name tags were removed. Most pictures included a tag with the current place name, making that tag the most important one for that area. However, its inclusion in the word cloud is redundant, as the user already knows where he is or which area he is looking at on the map. These multilingual place names were filtered out with the use of Wikipedia and Wikidata.

These techniques made the resulting word clouds much clearer, but they still contained some errors. The most common errors were due to bad translations or joined tags that are normally written with a space in between (e.g. "domkircheststephan"). This is a common problem with social media tags. The emergence of tags relating to the name or company of the photographer is another problem that occurs within the word clouds of smaller clusters. This spatial clustering of data visualizes the AOIs for each region and its general description through the eyes of the crowd. The AOIs often coincide with landmarks and popular areas of each city. As a next step, we investigated if there were differences in extracted AOIs when comparing people from different nationalities.

5.3.1 Tourim Interest Analysis

Only a fraction (32%) of the Flickr users provided information about their home location in their user profiles, limiting the available data for some countries of

origin. To classify the other users, a home determination method was developed based on [28]. The method considers all the posts created by each user and the country in which he has the most pictures is considered as a potential country of residence. If the time span between the first and the last post was greater than six months, the user was classified as a resident of that country. This algorithm was validated on the fraction of users who supplied information about their country of residence. This information was first preprocessed with Geonames to determine the English name of the city or country provided by the user. The developed algorithm achieved a precision of 0.87, a recall of 0.76, and an F_1 score of 0.81.

Kernel density estimation (KDE) [29] was selected as a visualization tool to generate continuous raster surfaces from the points. These surfaces are heatmaps representing areas with varying picture densities. KDE was chosen as most users were already familiar with the concept of heatmaps and it was immediately clear where the hotspots were located. Each heatmap shows the unique footprint of visitors from a certain country of origin. These heatmaps can be compared for different countries, to analyze the differences in tourism interest. Figure 5.4 shows the footprints of visitors from France, Japan, and the USA in Vienna. We see that the most popular areas of interest (the most popular tourist attractions) are shared and that larger differences occur in the less popular areas.

Figure 5.4: Footprints of visitors from France, Japan, and the USA in Vienna.

When looking at the generated tags for different nationalities, many tags were universal and widely used in the same locations. The most common tags were the more generic ones such as architecture, church, and travel. Between some nationalities, there were major differences in the areas or topics of interest. Figure 5.5 shows the word cloud for all points in Belgium from Dutch and English tourists. All of the points were included because the data is rather limited for specific regions of Belgium. It is clear that the interest of Dutch visitors, or at least those who posted on Flickr, was more focused on leisure activities (tomorrowland, motorsport) whereas the English tourists were more focused on traditional tourism. Tags related to the First World War (passchendaele, thegreatwar, memorial) show up in the data for English tourists, as Belgium (especially Ypres) is often visited to commemorate the Great War and its casualties.

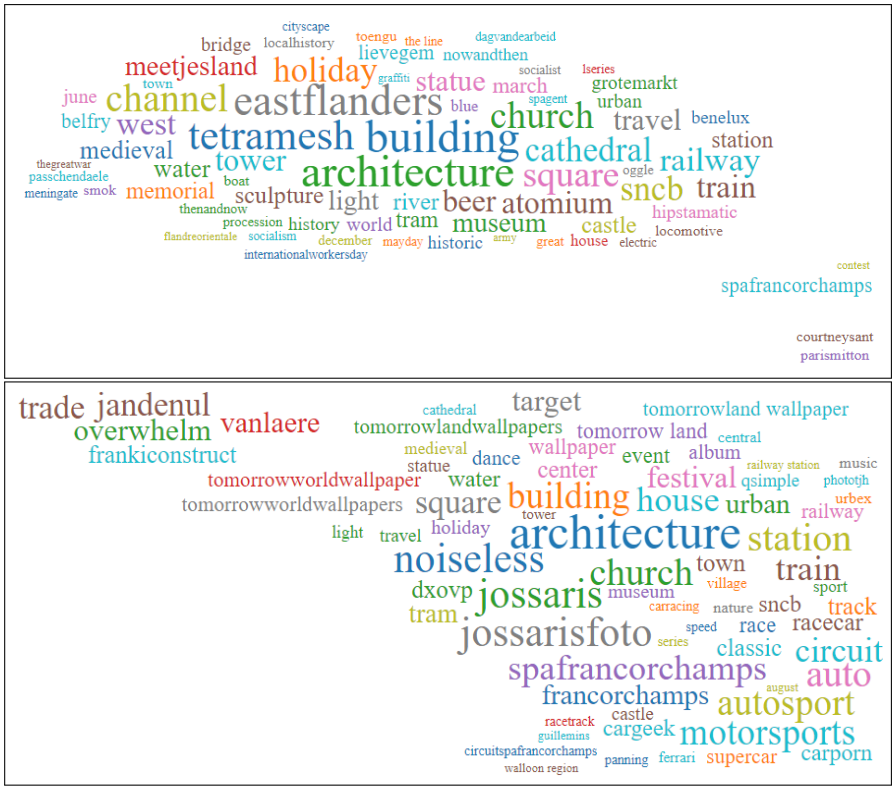


Figure 5.5: Word cloud of all points located in Belgium for English (top) and Dutch (bottom) tourists. Words are positioned relative to the user's location, which is Brussels for both word clouds.

Currently, the AOIs situated in Belgium and Austria were extracted at three different scales and their tags have been preprocessed and clustered. These clusters were visualized in an interactive map, where the word cloud of each cluster was shown when it was clicked. The current prototype is live and we have made the dataset publicly available. As suggested by [30], the word cloud algorithm was adjusted based on the location of each tag. The positions of the words on the word cloud corresponded to the location from where they were extracted, relative to the current user position. Both word clouds in Figure 5.5 were constructed with Brussels as the user's location. The tag Passchendaele is located on the left side (west) and Francorchamps is grouped with motorsport-related tags on the bottom-right (southeast). This visualization offers the benefit that it often groups related tags from the same place together, at the cost of introducing additional whitespace.

5.4 Conclusion

In this chapter, we have used NLP techniques to process and analyze social media data. Our pipeline successfully extracted and processed millions of tweets related to natural disasters and environmental topics. Such a pipeline should preferably include multilingual support to achieve better global coverage. Our initial tests show that there are spatiotemporal correlations between the occurrence of wildfires and the corresponding tweets. However, our current methods are not detailed enough to perform a thorough analysis of the immediate and long-term effects of these wildfires on global tweet behavior. Additionally, many of the collected tweets were not relevant to the queried topic but simply contained the same keywords or hashtags. We also showed that basic sentiment analysis models often fail to predict the correct sentiment and do not add much value when aggregated over large regions. Stance detection models can solve this issue, as our initial results showed good performance in determining the stance concerning nuclear energy. We plan to expand our dataset, label a larger number of tweets, and fine-tune state-of-the-art NLP models to gain further insights into the impact of environmental topics on Twitter. Our case study on Flickr data showed how NLP can be used to automatically uncover tourism hotspots. Furthermore, by analyzing the country of origin, we were able to uncover spatiotemporal differences between tourism hotspots.

References

[1] T. Kuemmerle, J. O. Kaplan, A. V. Prishchepov, I. Rylsky, O. Chaskovskyy, V. S. Tikunov, and D. Müller. *Forest transitions in Eastern Europe and their effects on carbon budgets*. Global Change Biology, 21(8):3049–3061, 2015.

[2] V. Tikunov, Y. Chereshnya, M. Gribok, and V. Yablokov. *Assesment of Russian regions in terms of the air pollution level*. Vestnik Moskovskogo Universiteta, Seriya 5: Geografiya, 2017:43–48, 01 2017.

[3] F. Horita, L. Degrossi, L. F. Assis, A. Zipf, and J. De Albuquerque. *The use of Volunteered Geographic Information and Crowdsourcing in Disaster Management: a Systematic Literature Review*. volume 5, 06 2013.

[4] T. Simon, A. Goldberg, and B. Adini. *Socializing in emergencies—A review of the use of social media in emergency situations*. International journal of information management, 35(5):609–619, 2015.

[5] J. P. de Albuquerque, M. Eckle, B. Herfort, and A. Zipf. *Crowdsourcing geographic information for disaster management and improving urban resilience: an overview of recent developments and lessons learned*. European handbook of crowdsourced geographic information, pages 309–321, 2016.

[6] E. Cambria and B. White. *Jumping NLP curves: A review of natural language processing research*. IEEE Computational intelligence magazine, 9(2):48–57, 2014.

[7] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry. *Semeval-2016 task 6: Detecting stance in tweets*. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pages 31–41, 2016.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805, 2018.

[9] S. Muralidharan, L. Rasmussen, D. Patterson, and J.-H. Shin. *Hope for Haiti: An analysis of Facebook and Twitter usage during the earthquake relief efforts*. Public relations review, 37(2):175–177, 2011.

[10] S. V. Ukkusuri, X. Zhan, A. M. Sadri, and Q. Ye. *Use of social media data to explore crisis informatics: study of 2013 Oklahoma Tornado*. Transportation Research Record, 2459(1):110–118, 2014.

[11] A. Acar and Y. Muraki. *Twitter for crisis communication: lessons learned from Japan's tsunami disaster*. International journal of web based communities, 7(3):392–402, 2011.

[12] V. K. Neppalli, C. Caragea, A. Squicciarini, A. Tapia, and S. Stehle. *Sentiment analysis during Hurricane Sandy in emergency response*. International journal of disaster risk reduction, 21:213–222, 2017.

[13] W. Maharani. *Sentiment analysis during Jakarta flood for emergency responses and situational awareness in disaster management using BERT*. In 2020 8th International Conference on Information and Communication Technology (ICoICT), pages 1–5. IEEE, 2020.

[14] G. Rybarczyk, S. Banerjee, M. D. Starking-Szymanski, and R. R. Shaker. *Travel and us: the impact of mode share on sentiment using geo-social media and GIS*. Journal of Location Based Services, 12(1):40–62, 2018. Available from: https://doi.org/10.1080/17489725.2018.1468039, arXiv:https://doi.org/10.1080/17489725.2018.1468039, doi:10.1080/17489725.2018.1468039.

[15] A. P. Kirilenko, T. Molodtsova, and S. O. Stepchenkova. *People as sensors: Mass media and local temperature influence climate change discussion on Twitter*. Global Environmental Change, 30:92–100, 2015.

[16] M. R. Sisco, V. Bosetti, and E. U. Weber. *When do extreme weather events generate attention to climate change?* Climatic change, 143(1):227–241, 2017.

[17] A. Reyes-Menendez, J. R. Saura, and C. Alvarez-Alonso. *Understanding #WorldEnvironmentDay User Opinions in Twitter: A Topic-Based Sentiment Analysis Approach*. International Journal of Environmental Research and Public Health, 15(11), 2018. Available from: https://www.mdpi.com/1660-4601/15/11/2537, doi:10.3390/ijerph15112537.

[18] E. Zotova, R. Agerri, and G. Rigau. *Semi-automatic generation of multilingual datasets for stance detection in Twitter*. Expert Systems with Applications, 170:114547, 2021. Available from: https://www.sciencedirect.com/science/article/pii/S095741742031191X, doi:https://doi.org/10.1016/j.eswa.2020.114547.

[19] B. Zhang, D. Ding, and L. Jing. *How would Stance Detection Techniques Evolve after the Launch of ChatGPT?*, 2023. arXiv:2212.14548.

[20] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. *Emergent Abilities of Large Language Models*, 2022. arXiv:2206.07682.

[21] F. Gilardi, M. Alizadeh, and M. Kubli. *ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks*, 2023. arXiv:2303.15056.

[22] D.-H. Yang, L. M. Bilaver, O. Hayes, and R. Goerge. *Improving geocoding practices: evaluation of geocoding tools*. Journal of medical systems, 28(4):361–370, 2004.

[23] S. Loria et al. *textblob Documentation*. Release 0.15, 2(8), 2018.

[24] M. Tkachenko, M. Malyuk, N. Shevchenko, A. Holmanyuk, and N. Liubimov. *Label Studio: Data labeling software*. Open source software available from https://github. com/heartexlabs/label-studio, 2020.

[25] D. Q. Nguyen, T. Vu, and A. T. Nguyen. *BERTweet: A pre-trained language model for English Tweets*. arXiv preprint arXiv:2005.10200, 2020.

[26] M. Lai, A. T. Cignarella, D. I. Hernández Farías, C. Bosco, V. Patti, and P. Rosso. *Multilingual stance detection in social media political debates*. Computer Speech & Language, 63:101075, 2020. Available from: https://www.sciencedirect.com/science/article/pii/S0885230820300085, doi:https://doi.org/10.1016/j.csl.2020.101075.

[27] R. J. Campello, D. Moulavi, and J. Sander. *Density-based clustering based on hierarchical density estimates*. In Pacific-Asia conference on knowledge discovery and data mining, pages 160–172. Springer, 2013.

[28] I. Bojic, E. Massaro, A. Belyi, S. Sobolevsky, and C. Ratti. *Choosing the right home location definition method for the given dataset*. In Social Informatics: 7th International Conference, SocInfo 2015, Beijing, China, December 9-12, 2015, Proceedings 7, pages 194–208. Springer, 2015.

[29] C. Grothe and J. Schaab. *Automated footprint generation from geotags with kernel density estimation and support vector machines*. Spatial Cognition & Computation, 9(3):195–211, 2009.

[30] B. Tessem, S. Bjørnestad, W. Chen, and L. Nyre. *Word cloud visualisation of locative information*. Journal of location Based services, 9(4):254–272, 2015.

6
Conclusion

"If you are not willing to be a fool, you can't become a master"

– Jordan Peterson

This final chapter gives an overview of the main findings of the performed research and topics discussed. The dissertation finishes with some ideas and directions for future work.

This dissertation presented our research on using AI-based methods to enrich digital collections. We've shown how these methods can be used in practice, on a variety of collections and data.

First, **we've shown how (semi)-automated methods can assist the digitization and labeling workflows.** Instead of manually preprocessing or annotating digitized objects, data-driven methods can automatically perform these tasks. The manual labor is then shifted towards validation instead of annotation, which is less complex and time-intensive. Semi-automatic labeling workflows provide the most benefit for complex annotations like pixel masks. We've shown how to effectively implement such pipelines, for digitized herbarium specimens, raster maps, and face recognition.

For digitized photographs, accurate open-source segmentation and retrieval models are already available. These can be used without any fine-tuning and

enrich the photographs by detecting a diverse set of common objects. **State-of-the-art multimodal transformers can be used to further enrich the collection with auto-generated captions and enable querying of the collection via natural language.** Furthermore, such similarity-based approaches can be used to find rare objects or geolocate archive photographs.

Using open-source facial recognition models, we have presented a generic pipeline to recognize persons of interest in archive photo collections. After semi-automatically constructing a reference dataset of 6075 unique persons of interest, our pipeline successfully recognized over 62,000 detected faces with a precision of 0.936. Frequently occurring persons, that were not initially part of the reference set, were later identified via clustering of the face embeddings. To validate these person predictions, an interactive labeling tool was developed. The tool collected over 180,000 labels and greatly sped up the validation process. **We conclude that facial recognition models can be applied accurately and at scale on archive photo collections.**

Our research on raster maps has resulted in an automated geolocation pipeline, based on text recognition and geocoding of the visible toponyms. The pipeline is generic and accurate, as long as the map contains enough toponyms. Using associated vector data, we were able to segment the roads on topographic maps with an IoU of 0.804. We then combined both methods on a dataset of walking and cycling maps, to predict the route GPS coordinates. However, additional research is needed to generalize these methods to historical (handwritten) maps.

Regarding herbaria, our research has optimized the digitization process and provided a way to automatically validate specimen names using the text labels. Furthermore, we created a novel herbarium instance segmentation dataset, containing pixel masks of the specimens and other objects on the sheets. Using this dataset, **we have evaluated several state-of-the-art segmentation approaches and have shown that these can accurately segment an entire herbaria sheet with an mAP score of 78.9.** By reformulating the problem as a panoptic segmentation task, we can combine the strengths of binary plant segmentation via Unet++ and object segmentation via YOLOv8. While our results are promising, additional research is necessary to scale it to the millions of publicly available herbarium sheets.

We've also demonstrated how to collect, process, and analyze large social media datasets. We've collected and geolocated over 15 million tweets, related to natural disasters and environmental topics. We've shown that simple sentiment analysis models do not suffice and created a small stance detection dataset. **By retraining a BERTweet model on this dataset, we achieved an F_{avg} score of 0.67.** We expect that this score can be improved significantly by labeling additional data. Furthermore, we've shown how to automatically determine tourism

hotspots and analyze these geospatially through a case study on a Flickr dataset.

There is generally no single solution to processing digital archives because the models and techniques will vary greatly depending on the context and requirements. However, we can still formulate a general pipeline that can be used to efficiently process large collections:

1. **Metadata generation**: Begin by employing pretrained AI models to generate (noisy) metadata. This includes image and person similarity, text recognition, object detection, and other relevant metadata.

2. **Annotation**: Use a semi-automated approach to quickly validate the noisy predictions or use domain adaptation techniques to reuse existing labeled datasets. Use these methods to create a ground truth dataset.

3. **Linking**: Enhance and validate the generated metadata by linking it with other data sources (e.g. pairing text recognition with geocoders, linking with Wikidata, etc.).

4. **Optimization**: Improve prediction accuracy by extending, fine-tuning, or combining multiple AI models tailored to the specific needs of the collection.

5. **Evaluation**: Assess the accuracy of your methods using the annotated dataset. Select the best method or model and apply it to the full collection.

6. **Visualization**: Aggregate and visualize the results for large-scale analyses. Implement interactive demonstrators for intuitive visualization and exploration of the collection.

7. **Dissemination**: Publish the models, code, and results to enable more efficient future research.

In summary, we have demonstrated how AI-based and data-driven techniques can assist in creating, maintaining, and analyzing digital archives. Implementing such techniques during digitization, processing, and visualization drastically reduces the amount of manual labor required and provides additional metadata, increasing the accessibility of the collection. However, there are still some challenges in processing historical collections without labeled data.

6.1 Future Work

One of the main challenges remaining is increasing the efficiency of data labeling. New foundation models like Segment Anything [1] can be used to quickly label complex objects, from points or bounding boxes. With some automated preprocessing, initial bounding box estimates could be generated, and then further fine-tuned via Segment Anything. However, this model was trained on photographs, therefore to further optimize the labeling process, it needs to be fine-tuned for new types of images, like herbaria or raster maps. Besides labeling additional data, an aggregation and standardization of available datasets would also be useful. For both herbaria and raster maps, some labeled datasets exist, but these are not uniform. The datasets will need to be merged and the missing objects need to be annotated in order to combine them.

Additionally, more complex augmentation methods can be used to further increase the variety of available data. These can range from copy-paste instance augmentation methods [2] to generative or domain adaptation methods. Specifically for raster maps, domain adaptation and synthetic data generation techniques could greatly increase the models' accuracy. You can use domain adaptation to generate weakly supervised labels, to train segmentation models [3]. For synthetic data generation, you could use open-source vector data from OpenStreetMap and use it to render map tiles in different styles. Then, to process a dataset of historical map sheets, like the Ferraris map[1], style transfer methods can be used. These will change the style of the OpenStreetMap tiles to a historical one. Previous work already used such an approach to generate additional synthetic text labels on raster maps, which improved OCR accuracy [4]. Going one step further, you could use a multitude of historical map collections as target styles and then retrain Segment Anything, to develop a generic historical map segmentation model.

For textual data, additional use of large language models like GPT-4 [5] could prove useful to automatically label additional stance detection datasets in a zero-shot way. These labels could then be quickly validated manually and used to retrain a supervised model. It is probably not cost-efficient to use GPT-4 or other LLMs to make predictions on millions of tweets. However, for smaller datasets, such models will likely achieve good results, with no fine-tuning. Creating or fine-tuning such models for specific domains or contexts would also be an interesting direction for future research. This way, these models would have a better understanding of the historical or domain-specific contexts, writing style, and nuances. You could then make them multimodal, combining two or more domains. For in-

[1] https://www.vlaanderen.be/datavindplaats/catalogus/ferraris-kaart-kabinetskaart-der-oostenrijkse-nederlanden-en-het-prinsbisdom-luik-1771-1778

stance, you could train the model on the text and images of historical newspapers. This allows querying of the texts and images via natural language, but also the answering of domain-specific research questions.

Developing more performant and smaller models will always be useful for digital archives, as it will reduce the cost of implementing such models. This is currently still a major bottleneck and the main reason why interactive, AI-based querying and filtering functionalities via natural language are not already integrated everywhere.

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. *Segment anything*. arXiv preprint arXiv:2304.02643, 2023. Available from: https://arxiv.org/abs/2304.02643.

[2] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph. *Simple copy-paste is a strong data augmentation method for instance segmentation*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2918–2928, 2021.

[3] S. Wu, K. Schindler, M. Heitzler, and L. Hurni. *Domain adaptation in segmenting historical maps: A weakly supervised approach through spatial co-occurrence*. ISPRS Journal of Photogrammetry and Remote Sensing, 197:199–211, 2023.

[4] Z. Li, R. Guan, Q. Yu, Y.-Y. Chiang, and C. A. Knoblock. *Synthetic map generation to provide unlimited training data for historical map text detection*. In Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, pages 17–26, 2021.

[5] OpenAI. *GPT-4 Technical Report*, 2023. arXiv:2303.08774.