

# Learning Image Similarities from Forensic Evidence

DISSERTATION

zur Erlangung des akademischen Grades

**Doktor der Technischen Wissenschaften**

eingereicht von

**Manuel Keglevic**

Matrikelnummer 0625732

an der Fakultät für Informatik  
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Diese Dissertation haben begutachtet:

---

Dimosthenis Karatzas

---

Andreas Maier

Wien, 23. September 2022

---

Manuel Keglevic





# Learning Image Similarities from Forensic Evidence

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

**Doktor der Technischen Wissenschaften**

by

**Manuel Keglevic**

Registration Number 0625732

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

The dissertation has been reviewed by:

---

Dimosthenis Karatzas

---

Andreas Maier

Vienna, 23<sup>rd</sup> September, 2022

---

Manuel Keglevic



# Erklärung zur Verfassung der Arbeit

Manuel Keglevic

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 23. September 2022

---

Manuel Keglevic



# Acknowledgements

I want to thank my supervisor Robert Sablatnig and all my colleagues at the CVL for their ongoing support. You helped me every time I got stuck with a particularly tricky problem and generally let me ask all the questions. The list is very long, so I will only name Florian and Simon. Thank you, Florian, for reading all those words! Of course, this work would not have been possible without the outstanding support from the lovely people from the BK who helped me understand forensics a little bit better. Your support allowed me to write a cohesive thesis, and it is nice to know that my research might positively impact the real world.

Finally, my family and friends. You know what you went through. Thank you. I might never be able to repay you, but I will try. Valeria, this is probably your PhD too.





# Kurzfassung

Die in dieser Arbeit behandelten forensischen Werkzeugspuren, Schuhspuren und handschriftlichen Dokumente sind für die Aufklärung von Verbrechen von entscheidender Bedeutung. Zum Beispiel können Schuhspuren, die bei Einbrüchen hinterlassen werden, ein Indiz dafür sein, dass ein mutmaßlicher Täter den Tatort betreten hat. Andererseits stellen gleichartige Schuhspuren an verschiedenen Tatorten ein Anzeichen auf eine mögliche Tatserie dar. Obwohl forensische Beweise akribisch gesammelt und digitalisiert werden, ist eine manuelle Suche nach übereinstimmenden Beweismitteln aus verschiedenen Fällen in einem Archiv mit Hunderten bis Tausenden von Bildern zeitaufwändig. Daher ist eine maschinelle Suche gewünscht, um forensische Experten bei dieser Aufgabe zu unterstützen. Dabei ist es das Ziel Bildähnlichkeiten automatisch zu analysieren, um den Experten die relevantesten Beweismittel zu einer Anfrage zu liefern. Diese Arbeit präsentiert daher eine Methodik für den automatischen Vergleich und die Suche forensischer Bilder. Im Gegensatz zu anderen Gebieten der automatischen Bildanalyse, wie z.B. der Objektklassifizierung, sind bei forensischen Bildern lokale Merkmale von entscheidender Bedeutung, da sie zur eindeutigen Identifizierung des Objekts oder der Person, die eine Spur am Tatort hinterlassen hat, führen können. Um ein Ähnlichkeitsmaß für diese lokalen Merkmale zu finden, das auf das jeweilige forensische Gebiet zugeschnitten ist, wird in der hier vorgestellten Methodik auf Metric Learning gesetzt. Dieser Ansatz ermöglicht einen effizienten Vergleich von lokalen Merkmalen in einem erlernten Embedding. Zusätzlich werden Methoden vorgestellt, die globale Abhängigkeiten der lokalen Merkmale beschreiben, mit dem Ziel verfügbare Daten effektiver nutzen und bereichsspezifische Einschränkungen modellieren zu können.

Eine Evaluierung der vorgestellten Methodik erfolgt mithilfe von Datensätzen aus den drei exemplarisch behandelten forensischen Bildmodalitäten. Darüber hinaus werden in dieser Arbeit zwei neue, öffentlich zugängliche Datensätze mit Werkzeugspuren und Schuhabdrücken vorgestellt, die explizit zum Training und zur Evaluierung von auf maschinellem Lernen basierenden Methoden entwickelt wurden. Dazu gehört auch eine umfassende Beschreibung der Aufnahmeabläufe zur effizienten Erfassung von etwa 7.000 forensischen Bildern. Da diese Arbeit alle Bereiche von der Erfassung und Annotierung einer großen Anzahl von forensischen Bildern, über die Entwicklung einer an den jeweiligen forensischen Bereich angepassten Methodik, bis zur Bereitstellung von interpretierbaren Ergebnissen für forensische Experten beschreibt, kann sie als Vorlage für eine effektivere Nutzung von physischen forensischen Beweisen mit Computer Vision Methoden dienen.



# Abstract

Forensic evidence, such as toolmarks, footwear impressions, and handwritten documents treated in this thesis, is crucial to solving criminal cases. For example, footwear impressions left behind during break-ins can place criminals at the scene of a crime and can be used to link different criminal cases together. Even though such forensic evidence is meticulously collected and digitalized, a manual search for matching evidence from different cases in an archive of hundreds to thousands of images is time-consuming.

In order to support forensic experts with this task, a retrieval system is desired that filters the results by relevancy using automatic analysis of image similarities. Therefore, this thesis presents a methodology for comparing and retrieving forensic images. In contrast to other image analysis tasks, like object classification, for forensic images, fine-grained local characteristics are crucial since they can uniquely identify the object or person that has left behind a trace on the crime scene. For toolmarks and footwear impressions, such characteristics occur, for example, due to damages or wear. Since they have the potential to yield the highest evidential strength, such individual characteristics are the most powerful during an examination. The proposed methodology facilitates metric learning to learn a similarity measure that is specific for each forensic domain addressed. This approach allows efficient comparison of local characteristics in a learned embedding space. In order to utilize the available data more effectively and provide a mechanism to enforce domain-specific constraints, methods for modeling the global context by combining local characteristics are presented.

The proposed methodology is evaluated using datasets from the three exemplary forensic image modalities addressed, i.e., toolmarks, footwear impressions, and handwritings. Further, two new publicly available datasets are presented, explicitly designed to train and evaluate learning-based methods for retrieving forensic toolmark images and footwear impressions. This includes a comprehensive description of the acquisition workflows developed to efficiently acquire about 7,000 forensic images.

Since this thesis describes the efficient acquisition and annotation of a large number of forensic images, the development of a methodology adapted to each forensic domain addressed, and an evaluation focused on providing interpretable results to forensic experts, it can be seen as a blueprint for utilize physical forensic evidence more effectively with computer vision methods.



# Contents

<b>Kurzfassung</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Contents</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Forensic Images . . . . .	2
1.2 Research Questions . . . . .	9
1.3 Contribution . . . . .	10
1.4 Thesis Structure . . . . .	13
<b>2 State of the Art</b>	<b>15</b>
2.1 Forensics Images . . . . .	15
2.2 Metric Learning . . . . .	29
2.3 Summary . . . . .	42
<b>3 Datasets</b>	<b>45</b>
3.1 FORMS Toolmarks . . . . .	47
3.2 Impress Footwear Impressions . . . . .	58
3.3 Summary . . . . .	66
<b>4 Methodology</b>	<b>67</b>
4.1 Toolmarks . . . . .	69
4.2 Writer Retrieval . . . . .	82
4.3 Footwear Impressions . . . . .	88
4.4 Summary . . . . .	92
<b>5 Evaluation</b>	<b>95</b>
5.1 Metrics . . . . .	96
5.2 Striated Toolmarks . . . . .	99
5.3 Toolmark Impressions . . . . .	109
5.4 Writer Retrieval . . . . .	117
5.5 Footwear Impressions . . . . .	125
	<b>xiii</b>

5.6 Summary . . . . .	134
<b>6 Conclusion</b>	<b>137</b>
<b>Bibliography</b>	<b>145</b>



# Introduction

Forensic evidence treated in this thesis are toolmarks, footwear impressions, and handwritten documents. Such forensic evidence is crucial to solving crimes. For example, footwear impressions left behind during break-ins can place criminals at the scene of a crime and can be used to link different criminal cases together. Even though forensic evidence is meticulously collected and photographed, a manual search for similarities of evidence from different cases in an archive of hundreds to thousands of images is time-consuming. In order to help the forensic experts detect linked cases in an extensive database of forensic images, a retrieval system is desired that filters the results by relevancy using automatic analysis of image similarities.

Even though forensic experts from different fields face the same problem of finding matching samples in extensive collections of physical evidence, the actual expertise needed is dependent on the type of forensic evidence. Furthermore, the techniques for acquiring digital images of such forensic evidence vary significantly between different forensic domains, from images photographed under the microscope with varying lighting conditions, images taken directly at crime scenes, images from lifters or 3D molds, and scanned images from sheets of paper. Therefore, three exemplary forensic domains (toolmarks, footwear impressions, and writer retrieval) were selected for this thesis to analyze which challenges are conceptionally similar and which difficulties are specific to the individual domain.

Generally speaking, for a given set of forensic images, four tasks can be identified utilizing an image similarity measure, namely verification, identification, retrieval, and classification, as shown in Figure 1.1 in the example of handwritten documents. These tasks mainly differ in the number of input samples and nature of the result [ALV11]. For example, for verification, only two samples have to be compared using a similarity measure, and the result is a binary decision, i.e., matching or non-matching. Likewise, the similarity of a sample to a limited number of classes has to be determined for classification. In contrast, identification and retrieval involve a comparison with all the samples in the

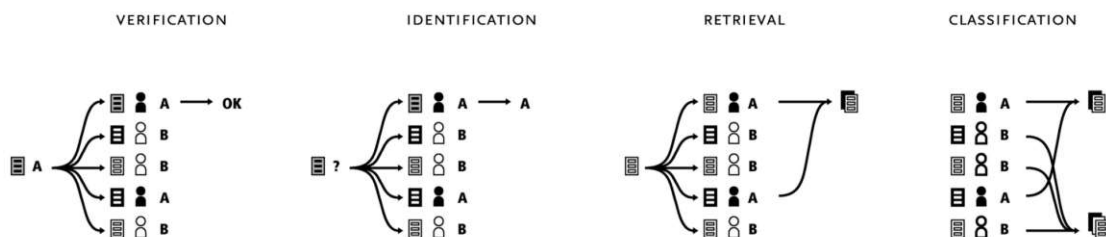


Figure 1.1: Processing tasks dealing with the similarity of handwritten document images [ALV11].

dataset to either identify who/what created the sample or retrieve all similar samples respectively. Consequently, the retrieval in databases that contain thousands of images involves computing the similarity of thousands of image pairs.

As this thesis focuses on retrieving forensic images, an efficient way to compute such similarities is required by finding compact representations of the distinguishing characteristics of forensic images.

## 1.1 Forensic Images

The research described in this thesis involved working with forensic experts from the Austrian Police and the Criminal Intelligence Service Austria (from now on referred to as Austrian Police) on three publicly funded security research projects. The domains selected, i.e., toolmarks, footwear impressions, and writer retrieval, were guided by the demand of the forensic experts in Austria for automatic comparison of such forensic evidence. The primary source for the forensic challenges presented in this section is the forensics experts' experience of the Austrian Police shared while working on these research projects.

### 1.1.1 Toolmarks

In case crimes are committed with the help of tools, toolmarks may be left behind. For instance, a common way of forced entry in Europe is lock-snapping. For this, a tool (for example, adjustable wrenches or locking pliers) is used to snap, i.e., break the lock. First, the part of the lock that sticks out of the door is gripped securely with the tool. Subsequently, in a back-and-forth motion, the tool is used to exert leverage on the mounting point of the lock that sits inside the door, which is the weak point of the cylinder lock and breaks if enough force is applied. Lock-snapping is visualized in Figure 1.2 on the left. The technique leaves an imprint of the tool used on the cylinder lock in the contact area, i.e., a toolmark unique to the tool used.

As an example, Figure 1.2 on the right shows multiple toolmarks of the same tool on a broken lock cylinder. By comparing two different toolmarks using a comparison



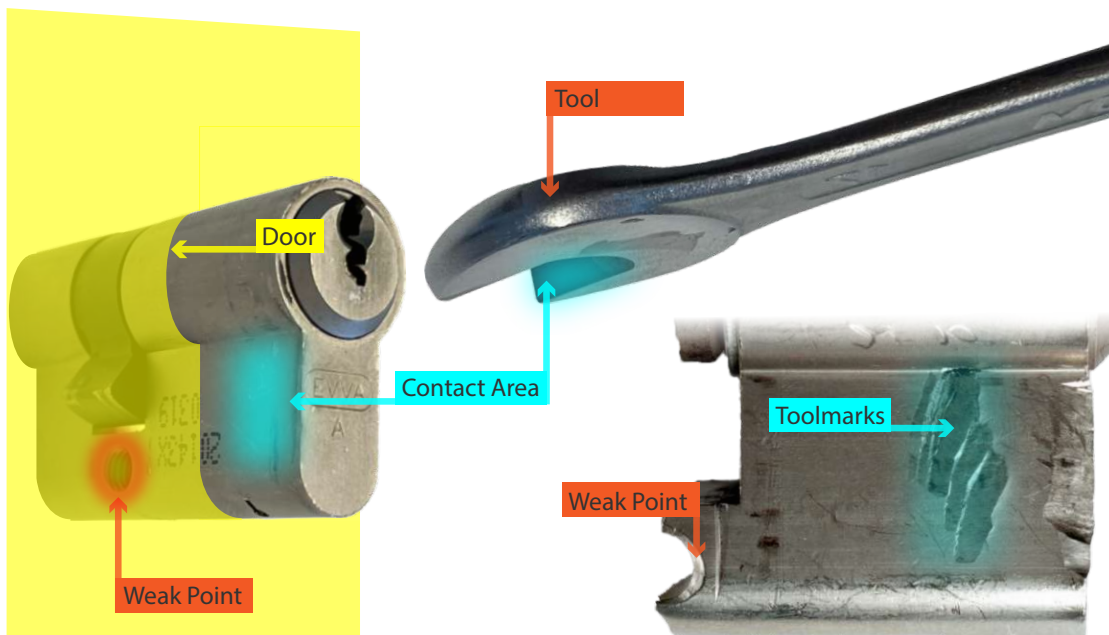


Figure 1.2: Illustration of lock-snapping. The cylinder lock snaps in half at the weak point, and toolmarks are left in areas where the tool is in contact with the lock. In this example, multiple toolmarks are visible on the snapped lock.

microscope, forensic experts can assess if the marks were made using the same tool. This analysis is conducted by searching for matching local characteristics. Subsequently, the positional relationship between such local matches is reviewed to ensure consistency in a global context. In Figure 1.3 and Figure 1.4 such a comparison is shown on impression toolmarks and striated toolmarks, respectively. The matching local characteristics are marked with matching colors in these examples. Similar to other forensic evidence found at crime scenes, toolmarks can either be used to confirm that a seized tool was used to commit a crime or to link multiple cases together and thereby significantly support the investigation of such offenses. Furthermore, the toolmarks found on these locks are crucial as evidence in the following court cases. Nevertheless, the manual examination and comparison of the toolmarks found is time-consuming due to the number of burglaries occurring every year (between 4,691 and 15,428 per year in Austria, according to the Ministry of the Interior [BMI22])

For the automated comparison and retrieval of toolmark images from the same tool, several challenges can be identified:

**Lighting** The visibility of the toolmark characteristics is heavily dependent on the lighting conditions. In order to allow discrimination of surface structure and toolmarks, lighting from oblique angles is needed to emphasize edges perpendicular to the lighting direction. Since the tools are not always used at the same angle, the opti-

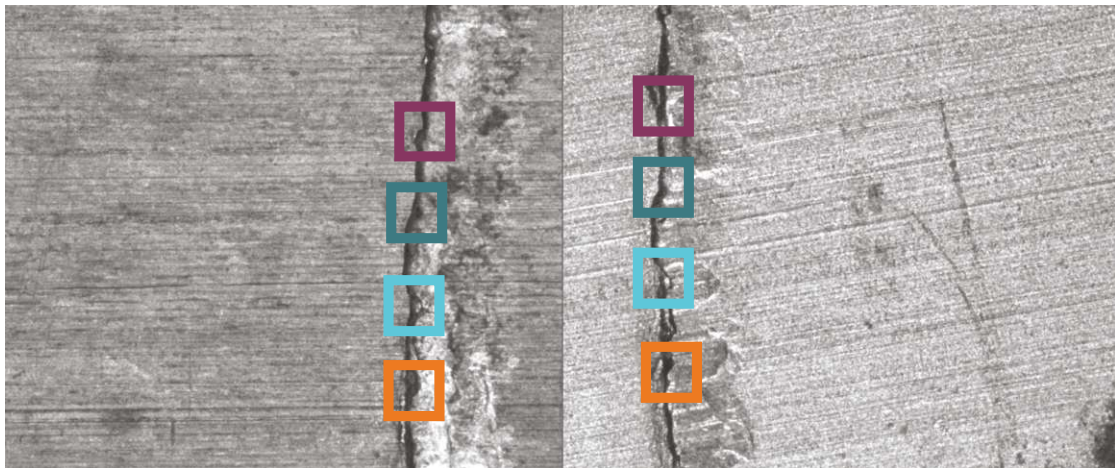


Figure 1.3: Matching local characteristics of two toolmark impressions made by the same tool.

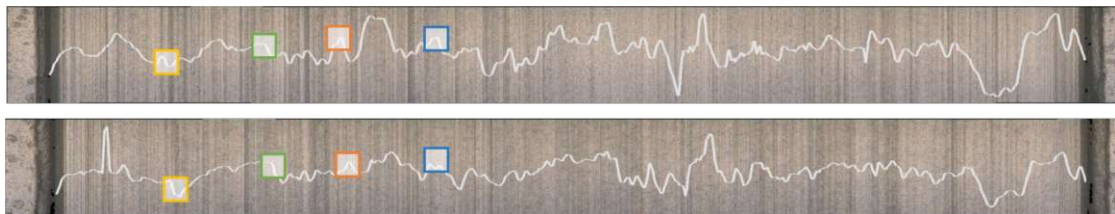


Figure 1.4: Matching local characteristics of two striated toolmarks made by the same screwdriver.

mal lighting direction varies between toolmarks. Consequently, images of different lighting directions must be compared as the lighting can not be standardized.

**Materials & Force Applied** Depending on the tool's and surface's materials and the force used, the depth and clarity of the toolmark impression vary. Small structures vanish in case the material is soft, and movement of the tool leads to blurred toolmarks.

**Individual Characteristics** The toolmark characteristics can be divided into characteristics shared by all tools of the same model and individual characteristics. Individual characteristics can either be unique characteristics due to the production process or damages due to wear (e.g., chipped-off pieces). During an examination, individual characteristics are the most powerful, as they have the potential to yield the highest evidential strength. Even though tools from the same assembly line share some characteristics, individual characteristics for each tool are still present. Nevertheless, shared characteristics can also have value, especially if the individual characteristics are absent.

**Acquisition System** Since toolmarks considered in this work are photographed using a microscope, technical parameters of the acquisition system, such as the camera's resolution and optics, the depth of field, and the microscope used, have to be considered. The resolution for instances defines if the fine-grained individual characteristics can still be distinguished. Similarly, a shallow depth of field leads to parts of the image being out of focus.

The main challenge for a similarity measure for toolmarks is detecting and comparing the characteristics of the tools while being invariant to varying parameters like angle of attack, substrate material, and lighting conditions. Such a similarity measure on the one hand has to be sensitive enough to distinguish the fine-grained individual characteristics and on the other hand robust enough to be invariant to changes in the aforementioned parameters.

### 1.1.2 Writer Retrieval

Writer retrieval is the task of retrieving document images with similar handwriting from a dataset. Experts then analyze this ranking, and thus new documents from the same writer can be found in an archive. Furthermore, if multiple documents from a single writer are found, connections between different historic manuscripts can be discovered. In the modern context, writer retrieval methods are used in forensics to analyze, for example, ransom or threat letters. It can link different letters and improve the chances of finding the author. In contrast to writer retrieval, writer identification is the task of finding the writer of a specific document. The writer has to be known in advance and their handwriting already analyzed for comparison. The procedure can be used to identify the writer of an unknown document in case several possible authors come into question.

Law enforcement agencies in Austria possess an extensive collection of handwritten documents. This collection includes, for example, documents belonging to open cases and reference samples from suspects and prisoners. However, these collections of documents can only be utilized to a limited extent since, for the identification of an unknown writer, all documents have to be compared manually by handwriting experts. By providing forensic experts with a writer retrieval system that allows for the search of similar handwritings, identifying unknown writers by handwriting experts can be expedited since only a small number of documents with similar handwritings have to be compared manually. Ideally, such a retrieval only requires digitalizing the handwritten documents using an image scanner to utilize the proposed system. This system also provides an effortless way to utilize existing databases with handwritten documents.

The handwriting style of people depends on different parameters like which pen is used or external influences such as distractions by something or someone. Thus, a person's writing exhibits slight changes from document to document, but also within a document itself, small variations occur. Figure 1.5 on the left shows a sample page from the CVL

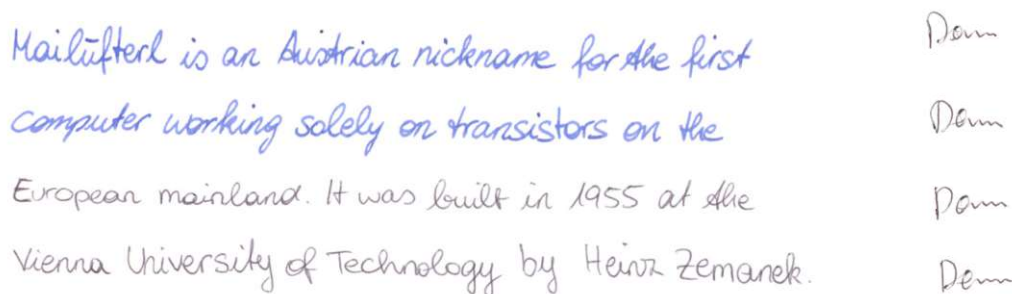


Figure 1.5: Samples from the CVL dataset. On the left, the writer used two different pens; therefore, the handwriting looks different. The image on the right shows crops where the German word “Dann” is written four times by the same writer and looks different each time.

Database [KFDS13] where the writer changed the pen during writing. The handwriting looks different at first glance, but by taking a detailed look at, for example, the word “the”, it can be seen that the same person wrote all four text lines. Figure 1.5 on the right shows another sample of the CVL Database with a text containing the German word “Dann” four times. The word is never written exactly the same way; slight variations occur in different characters. When applied to real-world samples, methods for writer identification and retrieval have to deal with variations like these. In the forensic context, writers may also intentionally modify their handwriting to prevent a comparison, e.g., common for threat letters or reference documents that suspects or inmates have been ordered to write.

In contrast to toolmarks and footwear impressions, the positional relationship between local characteristics does not define the writing style but the written text instead. The similarity of handwriting is defined by reoccurring local characteristics, like loops, that occur in different positions on the handwritten text. Figure 1.6 illustrates matching local characteristics with two handwritten texts by the same writer.

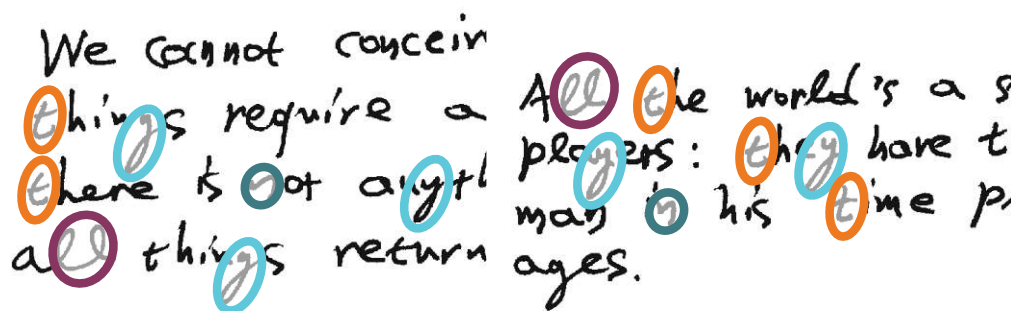


Figure 1.6: Matching local characteristics in two handwritten text by the same writer.

In contrast to toolmarks and footwear impressions, a person’s handwriting can even

heavily change when captured under laboratory conditions. The following challenges can be identified:

**People** In contrast to toolmarks and footwear impressions, for handwritings the goal is not to identify the object used to create the handwriting but the person creating it. Therefore, its appearance is not only influenced by the writer but also the pen used. Like the wear of a tool or shoe, a person's handwriting is not fixed and changes over time, i.e., the handwriting can change with the writer's age. However, the handwriting can even vary on the same page due to the writer's state of mind and external influences like distractions.

**Writer Invariant** The characteristics of handwriting are hidden under changing text, and thus, it is not enough to just compare local characteristics and their positions; instead, the distribution of strokes that make up a handwritten page has to be considered. The goal is to find the writer invariant, i.e., the reoccurring characteristics in the handwriting.

**Text** Automatic comparison of handwritings must ignore the actual text and only compare the writing style. As the written text is not always the same, specific words cannot be compared but just parts of the writing, like strokes, single characters, or frequently occurring character combinations like 'th'.

**Type and Alphabet** The handwriting style of a writer can vary with the type of handwriting, i.e., cursive, print, or modern cursive. That means that even the same sentence written in cursive may look different from that written in print. Such variations are even more significant when handwritten documents from different alphabets have to be compared, e.g., Latin and Greek.

### 1.1.3 Footwear Impressions

Since footwear impressions are frequently found at various crime scenes, they are a valuable source of evidence for criminal investigations. Especially for crimes committed mainly by serial offenders, like burglaries, comparing footwear impressions from different crime scenes allows investigators to link multiple cases together. Analogous to toolmarks, if a suspect is apprehended, the individual features of the footwear can prove that a specific shoe made a footwear impression, i.e., the suspect's shoe was at the scene of the crime. Forensic experts investigate the individual wear, damages, and manufacturing marks to prove this. Similar to toolmarks, this process is time-consuming and cumbersome.

However, footwear impressions are more common, and collections with footwear impressions by the Austrian Police contain thousands of images. Consequently, only a fraction of footwear impressions in these collections can be utilized without an automatic system that limits the number of necessary manual comparisons made by the forensic experts to the most similar footwear impressions. Furthermore, for time-sensitive cases it is crucial

to produce solid evidence to an investigator in time, e.g., when a suspect is being held in investigative custody and would otherwise have to be released.

In contrast to toolmarks, the model characteristics are the most prevalent for footwear impressions and can be used to narrow the search significantly without considering individual characteristics or wear. For example, Figure 1.7 shows a selection of common patterns found on shoe models. These patterns can also be employed to identify the shoe model using a reference database, which can aid investigators in searching for a suspect. However, as shown in Figure 1.8, these patterns are defined by reoccurring local characteristics, which requires a comparison of the distribution of the characteristics. Nevertheless, due to factors like wear, how the impressions are made, and the acquisition process, these local characteristics can change the appearance or not be visible in some parts of the footwear impression.

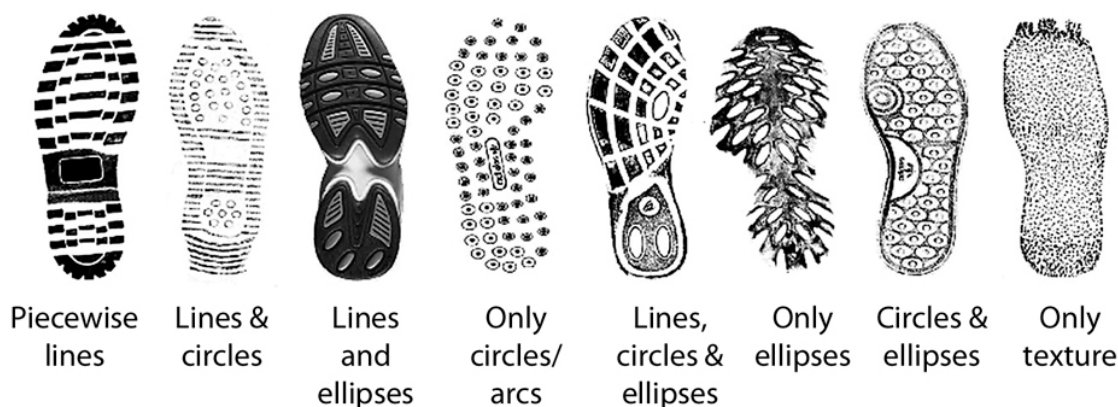


Figure 1.7: Multiple shoe models with distinct patterns.

To further narrow the search, individual characteristics can be compared to find multiple impressions of the same shoe. Similar to toolmarks, this includes characteristics created during production and blemishes due to wear. These individual characteristics can be compared similarly to toolmarks by finding multiple matching local characteristics and ensuring that their two-dimensional relationship matches.

For footwear impressions, the following domain-specific challenges can be identified:

**Quality Differences** The methodology has to consider the quality difference of footwear impressions collected at crime scenes and reference impressions of brand-new shoes created in a constrained environment. Footwear impressions found at crime scenes contain extensive noise and seldom show the whole shoe. Besides, multiple impressions from different shoes can overlap, and shoe soles change due to wear. Furthermore, images from shoe soles provide additional challenges, such as determining from a 2D image which part of the sole touches the floor, i.e., leaves an impression.

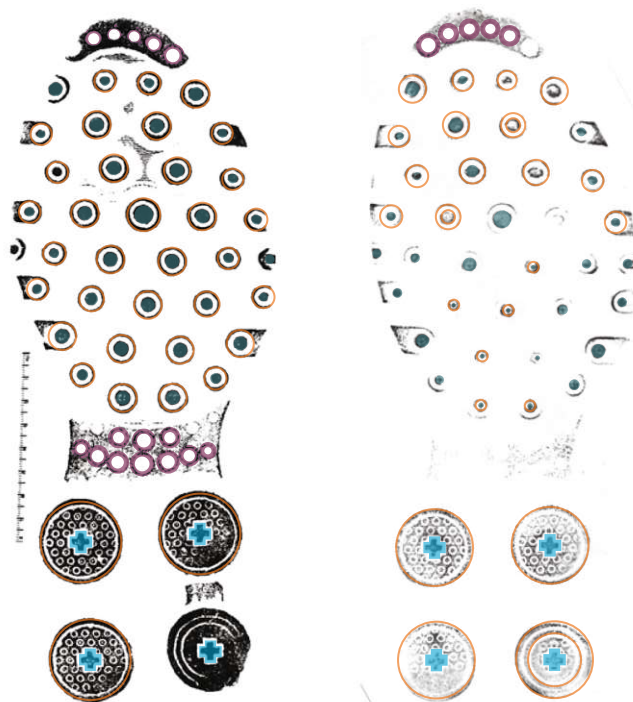


Figure 1.8: Reoccurring local characteristics highlighted on two footwear impressions made with the same shoe. The colors encode visually similar characteristics. Due to the acquisition process and how the impressions were made, these characteristics do not match in some parts.

**Model Characteristics** Even though a great variety of patterns are used to make shoe soles stand out, these patterns are not always unique to a specific model. The manufacturer often utilizes the same patterns for different models, and some types of shoes show very similar patterns across different manufacturers.

**Acquisition** There are different methods for securing footwear impressions; for instance, photographs are taken directly at the crime scene, gelatin or electrostatic lifters, and even 3D molds. This variety leads to significant differences in the captured impression images.

## 1.2 Research Questions

The main goal of this thesis is a methodology for retrieving forensic images of the presented domains to support the work of forensic experts. The research presented aims to answer the following research questions to achieve this goal:

### **Can deep learning allow a shared methodology for finding image similarities in different forensic domains?**

Even though a retrieval system can similarly assist experts of different forensic domains, the forensic images and the challenges involved vary significantly, as shown in the previous section. Therefore, this thesis examines the applicability of learning-based approaches for creating a similarity measure that ideally merely needs to be retrained for each modality while keeping most of the algorithm agnostic to the specific domains. Thus, this work explores how common approaches can be found for different forensic domains.

### **What are the shared concepts of the examined forensic image modalities?**

The three forensic image modalities (toolmarks, footwear impressions, and handwritings) are analyzed to define which challenges are conceptually similar and which are specific to the individual domain. In contrast to other image analysis tasks, like object classification, fine-grained local characteristics are as important as their global context for forensic images. First, this thesis investigates how these local characteristics from forensic images can be extracted to find matching local characteristics by computing their similarity. Subsequently, as presented in the previous section, the matching local characteristics must be placed into a global context to ensure that these local matches are consistent. Consequently, this thesis explores how this global context differs between the forensic domains considered. Furthermore, since the image modalities considered are diverse, with images photographed under the microscope with varying lighting conditions, images captured directly at crime scenes, images from lifters or 3D molds, and scanned images from sheets of paper, shared ways for efficiently acquiring, preprocessing, training and evaluating such data are explored.

### **How can the characteristics in the images be represented to enable an efficient search and comparison in databases?**

In order to assist forensic experts, the search in a database of forensic images needs to be fast. The computational effort for image retrieval is dependent on how efficiently a comparison with all images in a reference database can be performed. Typically, such a comparison involves evaluating a distance metric for the query image and each of the images in the reference database. Therefore, this thesis aims to discover ways to find a compact representation of the distinguishing characteristics of forensic images to allow the utilization of efficient distance metrics like the  $L_2$  distance.

## **1.3 Contribution**

The main contribution of this thesis is a methodology to compare forensic images automatically. In contrast to other image retrieval tasks, like object classification, fine-grained local characteristics are as important as their global context for forensic images. Therefore, the proposed methodology facilitates metric learning to learn a



similarity measure that is specific for each forensic domain addressed. This approach allows efficient comparison of local characteristics in a learned embedding space. In order to utilize the available data more effectively and provide a mechanism to enforce domain-specific constraints, methods for modeling the global context by combining local characteristics are presented. The proposed methodology is not one algorithm that fits all forensic domains. Nevertheless, it presents a core metric-learning-based approach that is adapted to handle local characteristics that are connected in a tightly constrained way (toolmarks), local characteristics that require compact modeling of their combined distribution (handwritings), and a combination of both (footwear impressions). The proposed methodology is developed and evaluated using the three exemplary forensic image modalities, i.e., toolmarks, footwear impressions, and handwritings. Furthermore, two new publicly available datasets were created explicitly designed to train and evaluate learning-based methods for forensic toolmark images and footwear impressions as part of this research.

Since this thesis describes all necessary steps from the efficient acquisition and annotation of a large number of forensic images, to the development of a methodology adapted to each forensic domain addressed, and an evaluation focused on providing interpretable results to forensic experts, it can be seen as a blueprint for how to utilize physical forensic evidence more effectively with computer vision methods. The following sections describe the domain-specific contributions in detail.

### 1.3.1 Toolmarks

Section 4.1 presents a metric-learning-based methodology for comparing striated toolmarks and impression toolmarks. To the best of my knowledge, the proposed methodology is the first that utilizes convolutional neural networks for comparing striated toolmarks. The evaluation in Section 5.2 demonstrates that the proposed TripNet can adapt to differences in angle of attack of  $15^\circ$  to  $60^\circ$ , which is the primary challenge for matching striated toolmarks of the NFI dataset [BKP<sup>+</sup>14]. Furthermore, the benefits of the proposed uncoupling of local characteristics from the global context for comparing striated toolmarks of unseen tools are demonstrated.

The proposed methodology is the first approach for automatically comparing toolmarks that has been developed and also tested on toolmark impressions, according to a survey by Baiker et al. [BHK<sup>+</sup>20]. The achieved performance demonstrates that with a probability of more than 70%, a matching toolmark can be found in case 20% of the images in a database of cylinder locks from real criminal cases are retrieved. The datasets developed for this evaluation, FORMS-Locks described in Section 3.1, is the first publicly available dataset with images from real criminal cases in the field of forensic toolmark comparison. It contains 3,046 images of 197 cylinders from 48 linked criminal cases captured using a comparison microscope in 11 different lighting conditions. Further, matching image regions in the toolmark images were manually annotated using an annotation tool developed as part of this work. Hence, in contrast to other datasets in this field, additionally to

the images and class labels, annotated local image similarities are provided to allow the evaluation of local image similarity measures in the context of forensic images.

The methodology for comparing striated toolmarks and impression toolmarks presented in Section 4.1, the evaluation described in Sections 5.2 and 5.3, as well as the FORMS toolmarks dataset presented in Section 3.1 are based on the following peer-reviewed publications:

- Manuel Keglevic and Robert Sablatnig. Learning a Similarity Measure for Striated Toolmarks using Convolutional Neural Networks. In *7th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pages 1–6. IET, 2016
- Manuel Keglevic and Robert Sablatnig. FORMS – Forensic Marks Search. In *Proceedings of the OAGM & ARW Joint Workshop 2017*, pages 111–112. Verlag der Technischen Universität Graz, 2017
- Manuel Keglevic and Robert Sablatnig. FORMS-Locks: A Dataset for the Evaluation of Similarity Measures for Forensic Toolmark Images. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1890–1897. IEEE, 2017
- Manuel Keglevic and Robert Sablatnig. Retrieval of striated toolmarks using convolutional neural networks. *IET Computer Vision*, 11(7):613–619, 2017
- Manuel Keglevic and Robert Sablatnig. Semi-Automatic Retrieval of Toolmark Images. In *Proceedings of the OAGM Workshop 2018*, pages 98–101. Verlag der Technischen Universität Graz, 2018

### Writer Retrieval

Section 4.2 presents a novel methodology for writer retrieval and identification based on learning an embedding representing the similarity of patches extracted from handwritten document images. Even though the proposed method does not match Christlein et al.’s [CBA15] performance using the VLAD encoding (88.0%) with a MAP of 86.1% on the ICDAR 2013 dataset, it demonstrates that metric learning can be utilized to learn the local characteristics of a writer’s writing style. Furthermore, a naive averaging of the embedding vectors achieves a MAP of 70.3%, which demonstrates that the learned embedding is able to capture the similarity of the local characteristics of the handwriting. In addition, a detailed evaluation of compactly encoding the distribution of such local characteristics using a Fischer Vector and VLAD is given. The methodology in Section 4.2 and evaluation for writer retrieval and identification in Section 5.4 are based on the following peer-reviewed publication:

- Manuel Keglevic, Stefan Fiel, and Robert Sablatnig. Learning Features for Writer Retrieval and Identification using Triplet CNNs. In *2018 16th International Con-*

ference on *Frontiers in Handwriting Recognition (ICFHR)*, pages 211–216. IEEE, 2018

### Footwear Impressions

The methodology for comparing footwear impressions presented in Section 4.3 and evaluated in Section 5.5 demonstrates the applicability of an end-to-end based approach for learning forensic image similarity. This is enabled by the Impress dataset described in Section 3.2, which was explicitly designed to allow the end-to-end training of neural networks. The proposed methodology allows an efficient search for similar impressions in a euclidean embedding. Furthermore, it is rotationally invariant and invariant (to some degree) to translations, scales, and aspect-ratio change. Thus, even though the proposed methodology cannot match the retrieval performance on the FID-300 dataset [KV16] of state-of-the-art approaches, like [KV16] and [KSRF19], it is more flexible and efficient since it does not require a time-consuming template-matching-like dense search for the best matching rotation and translation as these approaches. Additionally, the evaluation demonstrates that the proposed Impress dataset is diverse enough to train a flexible similarity measure that can handle samples from other datasets. The acquisition line designed for the Impress dataset allows an efficient collection of a mix of realistic but time-consuming and less realistic but less time-consuming impressions. The dataset is larger than any other publicly available footwear impression dataset, like the FID-300 [KV16] dataset and Richetelli et al.’s [RLL<sup>+</sup>17] high-resolution dataset, and contains 11 unique impressions for each shoe pair with over 4,000 images of 300 different pairs of shoes.

The presented methodology in Section 4.3, evaluation for footwear impression retrieval in Section 5.5, and description of the Impress dataset in Section 3.2 are based on the following peer-reviewed publication:

- Manuel Keglevic, Silvia Wilhelm, and Robert Sablatnig. Impress: A forensic footwear impression dataset. In *9th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pages 99–104. IET, 2019
- Manuel Keglevic and Robert Sablatnig. Impress: Forensic Footwear Impression Retrieval. In *Proceedings of the ARW & OAGM Workshop 2019*, pages 167–169. Verlag der Technischen Universität Graz, 2019

## 1.4 Thesis Structure

This thesis is structured as follows: Firstly, in Chapter 3 the datasets created in the course of this thesis are presented in detail. This chapter includes a comprehensive description of the acquisition workflows developed to acquire about 7,000 forensic images efficiently. Subsequently, Chapter 2 presents related work for automatically comparing toolmarks, handwritings, and footwear impressions, as well as general methods for computing image similarities using convolutional neural networks. In this section, an emphasis is placed on

metric-learning-based approaches. Afterward, in Chapter 4 the proposed methodology is described in detail, which includes the core metric-learning-based approach that is similar for all three forensic domains considered and the separate adaptations for each forensic domain. The methodology proposed is then evaluated in Chapter 5 on the datasets created in Chapter 3 and publicly available datasets. Finally, Chapter 6 summarizes the work presented in this thesis.

# State of the Art

In this chapter, first, related work for the automatic comparison of toolmarks, footwear impressions, and handwritings is presented in Section 2.1. Although prior work on utilizing automated approaches exists for all three forensic domains discussed, for toolmarks and footwear impressions, these papers focus more on acquisition methods, reproducibility, and statistical support for the work of forensic experts than on methods for automatically comparing such samples. In contrast to that, handwriting has been a focus of the computer vision community for a long time, and work on handwritten digits even led to the foundation of Convolutional Neural Networks (CNNs) [LBD<sup>+</sup>89]. Consequently, for the comparison of handwritings, this section focuses on computer vision approaches for writer retrieval and identification. Additionally, the available public datasets are presented for each domain. Special attention is given to the applicability of these datasets for learning-based training and evaluation strategies.

Subsequently, in Section 2.2 computer vision methodologies for image comparison are presented to allow for a detailed investigation of the potential improvements achieved by introducing such approaches into the forensic field. In particular deep-learning-based methods have shown to be adaptable for many fields and have replaced traditional methods as the current state of the art in computer vision applications, like classification, segmentation, object detection, and generative models [MBL20]. This section focuses mainly on methods based on deep metric learning since it allows for fast retrieval of similar images from collections containing thousands of reference images as each image is encoded as an embedding vector [MBL20].

## 2.1 Forensics Images

Similarly to identifying an individual's identity using physical, chemical or behavioral attributes of a person [JFR07], forensic examiners are tasked with identifying the marks or impressions left on crime scenes [Pet11] by objects. In case of biometrics, automated

searches for fingerprints became routine in 1983 by the FBI using the Automated Fingerprint Identification System (AFIS) [RB04]. Such systems utilize the anatomy of a fingerprint's local ridge structure which consists mainly of ridge endings and ridge bifurcations [RB04]. The relational data between these structures is then compared to produce 'candidate matches' that can further be investigated by human experts [RB04]. Other biometric fields involving the comparison of more complex structures have achieved renewed focus due to modern deep learning based methodologies [SKP15, PVZ15]. In biometrics, the forensic data is not limited to images but also indirect data like keystroke dynamics [TTY13] or chemical data as DNA [Pet11]. Nevertheless, the application of forensic images is not limited to the identification of a person or object, but also includes the automatic detection of alterations in digital images or videos [LL19, ANYE18, GD18], for example.

Like biometrics, this thesis focuses on finding similarities between forensic samples in order to facilitate the retrieval of likely matches. First, Section 2.1.1 presents related work on the diverse forensic field of toolmarks, which includes not only marks made by hand tools but also marks left in knife wounds. Secondly, in Section 2.1.3 state-of-the-art writer retrieval and identification methods are compared, and subsequently, in Section 2.1.2 automated approaches for the comparison of footwear impressions are presented. Finally, the differences between the state of the art of the presented forensic domains are discussed in Section 2.1.4.

### 2.1.1 Toolmarks

When a tool or object is placed against another object with enough force, the impression it leaves is defined as a tool mark [PC13]. Since toolmarks can be made with different tools or objects, the field of toolmarks is very diverse and contains multiple subdomains. For investigations concerning major criminal offenses like homicides, the forensic comparison of toolmarks can be very important [Pet11]. Examples for this are toolmarks found in wounds caused by knives [PC13], toolmarks found on padlocks that have been cut by a bolt cutter [FVNT15], cutter marks on thin (mm-sized) wires [HKB<sup>+</sup>14], screwdriver marks left when these tools are used to pry open a window [MWB<sup>+</sup>20, BKP<sup>+</sup>14] and marks left by wrenches on gas pipes [Pet11]. Furthermore, even though not considered tools [Pet11], firearms, are the primary cause of death in homicides, and their use leaves marks on the cartridge case and the bullets themselves similar to toolmarks [PC13]. More examples can be found in the books by Nicholas Petraco [Pet11], and Baldwin et al. [BBFR13]. The latter also gives a detailed introduction to how tools are manufactured and how wear affects the characteristics of the tools and the toolmarks, i.e., the features of toolmark patterns.

In the year 1993, the validity of comparative forensic examination of toolmarks was challenged in court in the United States by the "Daubert v. Merrell Dow Pharmaceuticals Inc." decision, which led to the forensic community focusing on obtaining statistical support for the notion of the *uniqueness* of toolmark patterns [SCE<sup>+</sup>15b], i.e., on showing that forensic toolmark examination is not an art, but science [Pet11]. The idea was to

show the existence of “measurable features with high degree of individuality” [BJJK10] to validate the identification of matching toolmarks as forensic evidence in court. Even though others [PTB11] pointed out that uniqueness is irrelevant and it is more important to use scientific methods to quantify the actual likelihood of an error, i.e., the likelihood of a mistake or misinterpretation by the forensic expert [PTB11], this led to works with the goal of showing the uniqueness of toolmarks that help understand how toolmark features look like, how they change between different tools, and how the manufacturing process, wear, substrate material and other factors influence the features.

For analyzing toolmarks automatically, first, 2D or 3D imaging of the forensic specimens is required. For ballistics, there have been publications on digital imaging since the 1970s [GBJ13] and in a survey from 2013, Gerules et al. [GBJ13] give an overview of 2D and 3D imaging techniques that are also relevant to toolmarks. In addition to laser scanning and mechanical probing, optical methods like scanning white light interferometry and confocal microscopy can be utilized to acquire a 3D surface topology of toolmarks [BPZ15]. For example, Baiker et al. [BPZ15] use an Alicona Infinite Focus Microscope with an optical system employing white light focus variation to capture toolmarks made by screwdrivers. For 2D imaging, toolmarks can be photographed using digital cameras at crime scenes or in the laboratory [Pet11] or using a forensic comparison microscope with a digital camera [BKP<sup>+</sup>14]. Similarly, Heikkinen et al. [HKB<sup>+</sup>14] utilize optical methods to obtain toolmark profiles of cuts made with different cutters, and they conclude that using 3D imaging, a manual matching of the corresponding profiles is possible since there are enough features visible even when wear is introduced.

Other than wear, Baiker et al. [BPZ15] show that the substrate material influences the quality of the toolmark, which can lead, for instance, to a more prominent replication of the surface structure of the tool. However, they notice that the difference is subtle and primarily for structures in the 5–10  $\mu\text{m}$  range and that wax is a better alternative to lead as a substrate material. Similarly, they show that toolmark similarity and variability depend on the angle of attack and the depth of the toolmark, with shallower toolmarks offering a better quality [BPZ15]. For their investigations, they used two similar types of slotted screwdrivers, which should have unique surfaces due to the manufacturing process [BPZ15]. Similar experiments were conducted by Puentes et al. [PC13] using three different knives and ribcages of 6 male cadavers to investigate the reliability of cut mark analysis in human costal cartilages. They show that class characteristics of the knives can reliably be identified using the striation patterns created by the knives. However, they note possible limitations due to the small number of individuals manually producing the cuts (two) and the small number of different knives used (three) in the experiments.

The “Daubert v. Merrell Dow Pharmaceuticals, Inc.” decision also led to the development of computer-based methods for the automatic comparative examination of toolmarks [SCE<sup>+</sup>15b]. Even though comparing toolmarks has proven to work in the field and Murdock et al. [MPT<sup>+</sup>17] argue that other errors in criminal cases are way more common than errors due to toolmark comparison, methods that have a known or

potential error rate are desired by the forensic community [MPT<sup>+</sup>17]. In contrast to DNA profiling, which allows for the calculation of the Random Match Probability (RMP), there is no RMP for toolmark comparison and other forensic evidence like fingerprints, shoe prints, or tire tracks [MPT<sup>+</sup>17]. For many publications in this field, the automatic comparison of toolmarks acts in this regard just as a means to prove that (manually) comparing toolmarks works and is meant to provide objectivity to the subjective field of toolmark comparison.

Traditionally, Consecutive Matching Stria (CMS) is used by examiners to determine if two striae marks match as a quantitative identification criterion [CTSV13]. By aligning the striation patterns of striated toolmarks, the number of consecutive matching stria is used to define if toolmarks are identified as a match by the expert [CTSV13]. Using 3D surface scans, Chu et al. [CTSV13] propose the automatic detection of striation of bullet surfaces by using a heuristic for detecting stria based on peaks and valleys in the Gaussian band-pass filtered cross-section profiles.

For the comparison of striated toolmarks, a variety of methodologies [BJJK10, BKP<sup>+</sup>14, BPZ15, BPG<sup>+</sup>16, CTSV13, CMK<sup>+</sup>10, PCDF<sup>+</sup>12] operate on 1D profiles extracted from either 2D images or 3D surface scans of the striated toolmarks. After preprocessing, similarity scores are commonly computed using either global [BKP<sup>+</sup>14, BPZ15, BPG<sup>+</sup>16] or local [CMK<sup>+</sup>10, BJJK10] cross-correlation. Bachrach et al. [BJJK10], for instance, propose the use of locally normalized squared distances, i.e., cross-correlation, as a similarity measure, which they call *relative distance*. This approach is also proposed by the National Institute of Standards and Technology (NIST) for comparing ballistic toolmarks [RCLJ15, SV00]. In contrast to computing a similarity measure, Petraco et al. [PCDF<sup>+</sup>12] propose a classification approach based on machine learning. In the first step, Principal Component Analysis (PCA) and Linear discriminant analysis (LDA) are used for dimensionality reduction of the input profiles. The identity of the tool, i.e., the class, is then predicted using Support Vector Machines (SVM).

Mattijssen et al. [MWB<sup>+</sup>20] investigate the reliability of forensic firearm examiners and compare their performance to an automated method that is based on [BKP<sup>+</sup>14]. In this work, similarly to the striated toolmarks treated by Baiker et al., they compute 1D profiles from 2D images and 3D surface scans of the striation pattern of the firing pin on the cartridge case of test shots from 200 Glock pistols. After frequency filtering and adjusting for scale and registration, they compute a similarity score using cross-correlation. For validation, they show 60 comparisons to 77 forensic firearm examiners. Their automated approach is better suited to identify known matches with a true positive rate of 94.7% compared to 93.2%. However, in their studies, forensic firearm examiners are more reliable at identifying known non-matches with a true negative rate of 81.0% vs. 77.3% for the automatic comparison of 3D surface scans. For the profiles computed from the 2D images, their method only reaches a true negative rate of 54.5%, which shows that their method does not work well with 2D image data. Nevertheless, they noticed that the results vary significantly among individuals, with a 95% confidence interval of [0.773, 0.847] for the true negative rate and [0.915, 0.949] for the true positive rate.



The common challenge for comparing striated toolmarks lies in detecting and comparing individual, class, and sub-class characteristics of the tools [BKP<sup>+</sup>14]. Further, parameters like angle of attack ( $\alpha$ ), substrate material, and axial rotation have a significant impact on toolmarks [BPZ15]. Baiker et al. [BKP<sup>+</sup>14] show that when comparing toolmarks with different  $\alpha$ , for differences of 30°, the error rate is more than an order of magnitude higher than for differences of 15°, i.e., the false discovery rate increases from 3.00% to 36.67%. Ekstrand et al. [EZG<sup>+</sup>14] create virtual toolmarks using a digitalized 3D version of the tool's tip and compare these to toolmarks on lead plates with a difference in  $\alpha$  of  $\pm 5$ –10° to circumvent these issues. They show that this approach can correctly identify matches with zero positives and two false negatives using the same cross-correlation-based approach as Baiker et al. [BKP<sup>+</sup>14]. However, this methodology fails to correctly match toolmarks that were created with differences over 25.3° in  $\alpha$  [SCE<sup>+</sup>15a]. To circumvent these shortcomings, Spotts et al. [SCE<sup>+</sup>15a] create virtual toolmarks at different  $\alpha$  to correctly identified all known matches of the physically created toolmarks without any false positives using the same methodology.

However, all approaches described above rely on striated toolmarks created under laboratory conditions with fixed angles of attack, constrained lighting conditions, high-resolution 3D surface scans, and hand-selected tools and surface materials. Baiker et al. [BHK<sup>+</sup>20] provide an extensive summary of the work done in this field in the years 2016-2019. In their survey, they found that although there has been extensive work on comparing striated toolmarks, the methodology described in Section 4 is the first publication on toolmark impressions.

## Datasets

Even though there have been several publications on the creation of 2D and 3D toolmark datasets by systematically creating toolmarks and using imaging techniques to digitize these marks, only the NFI Toolmark dataset, created by Baiker et al. [BKP<sup>+</sup>14], was made publicly available. Baiker et al. use 50 off-the-shelf screwdrivers of two different models in their work. A detailed description of the dataset is given in Section 5.2.1. Other publications in this field include Bachrach et al. [BJJK10], who use toolmarks made with 10 different screwdrivers of the same manufacturer and model number to examine the statistical distributions of similarity values, Petraco et al. [PCDF<sup>+</sup>12], who used 36 different screwdrivers, and Spotts et al. [SCE<sup>+</sup>15b], with 50 sequentially manufactured slip-joint pliers. For firearms, the NIST Ballistics Toolmark Research Database (NBTRD)<sup>1</sup>, provides reflectance microscopy images and three-dimensional surface topography.

Ekstrand et al. [EZG<sup>+</sup>14] create virtual toolmarks from both sides of six screwdriver tips using 3D optical profilometry and compare these virtual toolmarks to toolmarks on lead plates at three different angles of attack (45°, 60°, and 85°). Spotts et al. [SCE<sup>+</sup>15a] also experimented with creating virtual toolmarks by scanning six screwdriver tips at a

<sup>1</sup><https://tsapps.nist.gov/NRBTD/>

45° angle using an optical profilometer in 3D. The advantage of their method is that they can create toolmarks at multiple angles to circumvent the shortcomings of automatic comparison algorithms like [CMK<sup>+</sup>10] that fail to match toolmarks with angle differences over 25.3° correctly.

However, the toolmarks were created in constrained laboratory conditions in all those publications. The tools and surface materials were hand-selected, the toolmarks were made in a reproducible way using a fixed angle of attack, the lighting conditions were constrained, and the images or 3D surface scans are available in very high resolution; more than 400 pixels/mm as for instance in the case of the NFI Toolmark dataset created by Baiker et al. [BKP<sup>+</sup>14]. Therefore, assessing the real-world performance of the automatic comparison of toolmarks is not possible without a new dataset. Examples from this dataset are shown later on in Chapter 4.

### 2.1.2 Handwritings

Even though the authentication of a person by handwriting is very old (one of the first publications dates back to the 1920s), in the 2000s, the forensic community started to gain interest in automatic writer identification due to forensic applications like handwritten anthrax letters [KCS14, BS07]. Nevertheless, all state-of-the-art automatic writer identification and retrieval methods described in this section come from the computer vision community.

In 2001, Marti et al. [MMB01] proposed the analysis of the handwritten characters themselves by describing the slant and the heights of the different writing zones. Similarly, Bulacu et al. [BS07] propose to use different features like contour direction, contour-hinge, and direction co-occurrence. More recently, Jain and Doerman [JD13] extended this idea by proposing a Contour Gradient Descriptor.

Other methods calculate local features on the document image describing the neighborhood of specific points. Fiel and Sablatnig for example use SIFT features in [FS12] and [FS13] which describe the neighborhood of keypoints. Nicolaou et al. [NBLK15] use Local Binary Patterns, which are calculated for each pixel.

Deep learning methods, which have been used in digit recognition as one of the first applications [LBBH98], have also found their way back to the field of document image analysis, e.g., handwritten text recognition [SRTV16]. For writer retrieval, the feature distribution for local image regions computed using CNNs is used to describe a writer's handwriting. Examples of this are Chu and Srihari [CS14], Fiel and Sablatnig [FS15], Christlein et al. [CBA15], and Xing and Qiao [XQ16]. These methods train CNNs on a classification task and use the activations of one of the last fully connected layers of the network as a feature descriptor for each image patch and combine them afterward to generate a feature vector for the complete document image. As classification labels for training, writers are a natural choice to use, as done by Fiel and Sablatnig [FS15] for example. More recently, Christlein et al. [CGFM17] showed that instead, unsupervised clustering can be utilized to compute surrogate classes. Using a Resnet20 and patches

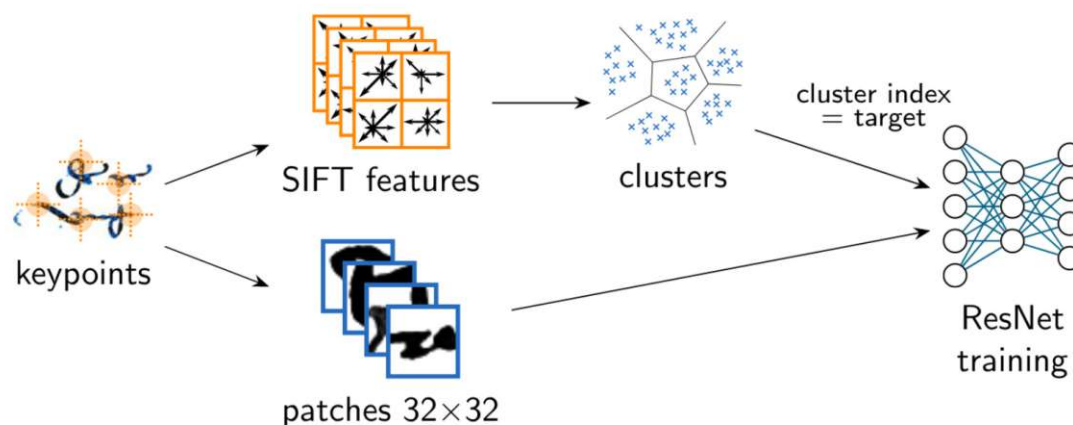


Figure 2.1: Writer retrieval approach using SIFT features and unsupervised clustering for ResNet training [CGFM17].

extracted from SIFT locations, they compute surrogate classes by clustering 500k randomly chosen SIFT descriptors from the training set using k-means. After filtering out descriptors between clusters, they use the cluster labels to train a Resnet20 network using a classification loss. The resulting 64-dimensional feature vectors from the penultimate pooling layer are then utilized to represent local features. Their approach is illustrated in Figure 2.1.

For aggregating and encoding these local features to allow a fast retrieval, earlier methods like [FS12], used a Bag-Of-Words (BOW) approach, for example. BOW utilizes a histogram of clustered features to encode the co-occurrences of visual words [FS12]. Fiel and Sablatnig [FS13] later improved their writer retrieval approach by using a Fischer Vector instead of BOW, which encodes the mean and variance of Gaussian Mixture Models fitted to the data. Likewise, Christlein et al. [CGFM17] propose VLAD [JPD<sup>+</sup>12] to encode local handwriting feature. In contrast to the Fischer Vector, it just encodes the residuals to the cluster centers, but it does not encode higher-order statistics. A detailed description of VLAD and the Fischer Vector can be found in Section 4.2.3. Nevertheless, Fiel and Sablatnig [FS15] showed that simply averaging local CNN features can also be utilized to aggregate the local features for writer retrieval. Rasoulzadeh et al. [RB21] proposed Generalized Max Pooling (GMP) for this aggregation to improve the encoding of clustered local features. By utilizing an optimization process, GMP seeks to maximize the similarity between the pooled representation and each local feature [MP14].

Christlein et al. [CSS<sup>+</sup>19] extended the idea of GMP as a trainable Deep GMP network layer. This layer allows them to train the model end-to-end with whole pages instead of just utilizing the local features generated from patches. This idea is extended by Wang et al. [WMC21] proposing an end-to-end trainable model that even includes a U-Net for binarization as a first block. The whole model, which utilizes the Texture Encoding Network [ZXD17] model Deep TEN to directly learn visual vocabularies, is

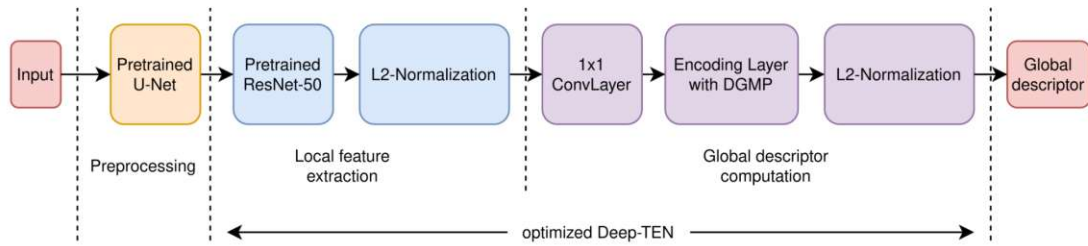


Figure 2.2: End-to-end trainable model for writer retrieval utilizing U-Net for binarization and Deep TEN for feature extraction and encoding [WMC21].

shown in Figure 2.2. Similar to the NetVLAD utilized by Rasoulzadeh et al. [RB21] Deep TEN uses a learnable encoding layer that calculates residuals to visual codewords and aggregates these residuals. Like NetVLAD a soft-assignment of the descriptors to codewords is employed to make the model differentiable and allow training via backpropagation [AGT<sup>+</sup>16, ZXD17]. However, in contrast to Deep TEN, NetVLAD has separate weights for learning the assignments and the codewords [AGT<sup>+</sup>16]. Both Rasoulzadeh et al. [RB21] and Wang et al. [WMC21] utilize a triplet margin loss, as described in the following Sections 2.2 and 4.1.2. However, the NetVLAD approach proposed by Rasoulzadeh et al. [RB21] is trained on  $32 \times 32$  patches, contrary to Wang et al. [WMC21] who use bigger random crops with a size of  $300 \times 300$  pixels.

On the ICDAR 2017 historical dataset [FKD<sup>+</sup>17], the Deep GMP [WMC21] approach achieves a top-1 accuracy of 71.2%. Utilizing Deep-TEN, slightly lower performance of 67.9% is achieved [WMC21]. Wang et al. [WMC21] also propose fine-tuning the U-Net used for preprocessing simultaneously with the rest of the network. However, their findings show that this does not improve the results and should be done as a separate preprocessing step [WMC21]. Rasoulzadeh et al. [RB21] did not evaluate their approach with the Historical-WI dataset. However, they achieve state-of-the-art performance with a MAP of 97.41% and 98.6% on the ICDAR 2013 and CVL datasets, respectively. For these results, they used NetVLAD in combination with a re-ranking that utilizes k-reciprocal nearest neighbors. This method, which has been proposed for re-ranking person re-identification results [ZFD17], compares the k-reciprocal nearest neighbors of the query and the gallery to improve the ranking of the retrieved results. Rasoulzadeh et al. [RB21] demonstrate that this re-ranking is especially efficient for the ICDAR 2013 dataset, where the performance is increased by 3.57% MAP. On this dataset, their method also clearly outperforms the approach by Christlein et al. [CM18] who use an Exemplar SVM to boost the performance of a VLAD encoding to achieve a MAP of 93.2% on this dataset. However, for the CVL dataset, the MAP is similar with 98.4%, and the top-1 accuracy of Christlein et al.'s approach [CM18] is slightly better. On the KHATT dataset Christlein et al. [CM18] still outperform other state-of-the-art approaches with a MAP of 98%, although only by 0.8 – 0.3%.

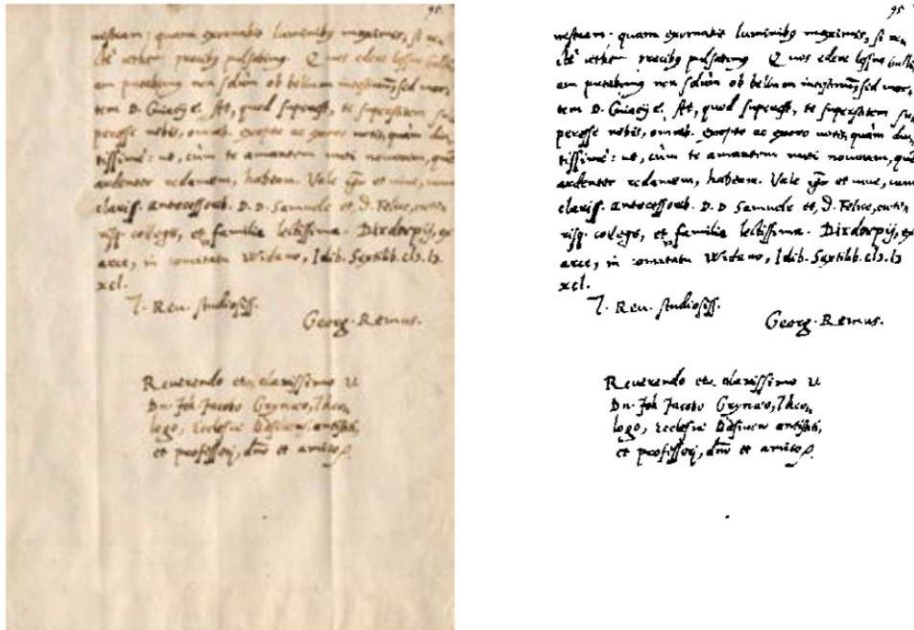


Figure 2.3: Example page from the ICDAR 2017 dataset which are provided as color images (left) and binarized (right) [FKD<sup>+</sup>17].

## Datasets

For writer retrieval, numerous dataset exists with images of modern and historical handwritten pages. Examples of such datasets with modern handwritten English pages are the CVL [KFDS13], ICDAR 2013 [LGSP13], and IAM [MB02] databases. In contrast to other datasets, the ICDAR 2013 database contains not only English but also Greek pages. Furthermore, since it was published as part of a competition, it contains separate training and testing sets and is therefore regularly used for evaluating training-based approaches, for instance, [RB21], and [CM18] described above. In contrast to the CVL and ICDAR 2013 database, which contain an even distribution of handwritten pages per writer, the IAM database contains only one page for approximately 350 writers [KFDS13]. Therefore, the IAM database requires special attention during training and evaluation and is, as such, not as well suited for evaluating learning-based approaches. The CVL and ICDAR 2013 datasets are described in detail in Section 5.4.1.

In contrast to these datasets, the ICDAR 2017 [FKD<sup>+</sup>17] and ICDAR 2019 [CNS<sup>+</sup>19] datasets contain historic handwritten pages. Even though these datasets contain thousands of images, e.g., 20,000 documents from 10,000 writers in the ICDAR 2019 dataset, the handwriting differs severely from modern handwriting and is therefore not considered in this thesis. Figure 2.3 shows an historic document from the ICDAR 2017 [FKD<sup>+</sup>17] as an example. Other datasets that contain only handwritings from other alphabets



Figure 2.4: Real crime scene footwear impression(s) collected using a gelatin foil lifter.

than Latin-script, like the KHATT database with handwritten Arabic text [MAA<sup>+</sup>12], are similarly not considered in this thesis since it focuses on German and English handwritings.

### 2.1.3 Footwear Impressions

Earlier approaches for the automated comparison of footwear impressions suggested the use of frequency analysis [dFR05, GBCN08] or local descriptors like Hu-Moments [AH08] and Scale-Invariant Feature Transform [SCBG07, SCB07, NBC<sup>+</sup>09]. However, as shown by Luostarinen and Lehmussola [LL14] in a survey paper published in 2014, footwear impressions from real criminal cases contain too much noise, and therefore even the best performing of these approaches, by Gueham et al. [GBCN08], and Nibouche et al. [NBC<sup>+</sup>09], fail at this task. Besides noise, other challenges are blurred, partial, and overlapping impressions which frequently occur at crime scenes. In Figure 2.4 this is shown in an example. The dust in the background leads to a very noisy image in which the foreground, i.e., the impression, cannot be separated from the background clearly. Furthermore, multiple impressions and blur make it hard to find one clean impression in the image. Nevertheless, for footwear impressions of the *Good* dataset, both, Gueham et al. [GBCN08], and Nibouche et al. [NBC<sup>+</sup>09] achieve almost 100% rank-1 performance, which drops to 85% for the *Bad* dataset. Both can handle rotations. However, since Gueham et al. [GBCN08] globally apply the Fourier-Mellin Transform, it does not work with partial impressions. Contrary, Nibouche et al. [NBC<sup>+</sup>09] use SIFT with RANSAC and can therefore handle partial impressions. Regardless, both approaches perform worse for footwear impressions from real criminal cases (no partials), where for a majority of the samples, 10%-60% of the database has to be searched to find a match [LL14].

To better handle impressions from real crime scenes, Wang et al. [WSYZ15] propose the combination of local descriptors and frequency analysis by using Wavelets and the Fourier Transform. Unfortunately, this requires a clean separation of the footwear impression

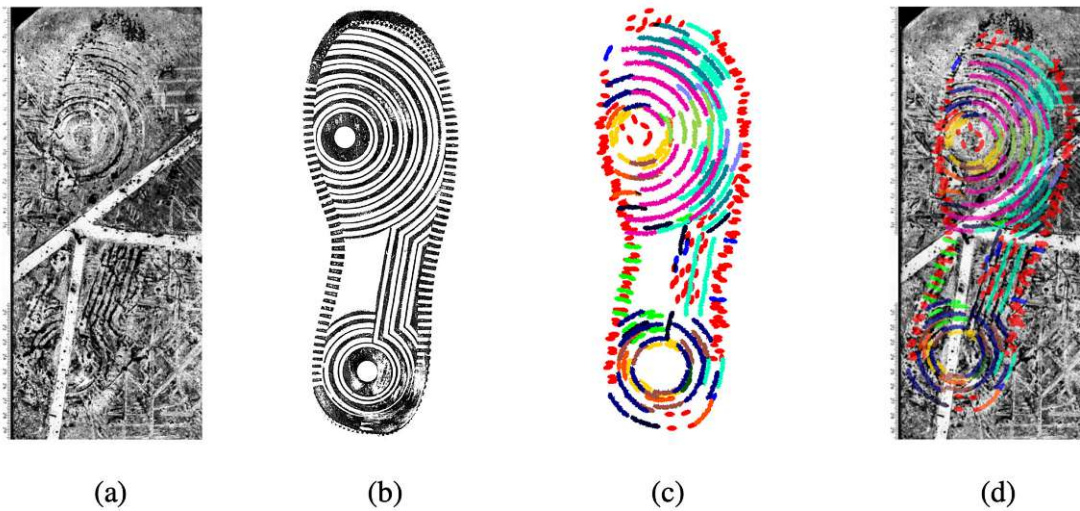


Figure 2.5: Representation of the footwear impressions as a composition of active basis models with a crime scene impression (a) and a reference image (b). The learned composition of active basis models (c) is spatially transformed to maximize the probability of a match (d) [KV16].

from the background as a preprocessing step, and they provide no comparable results on publicly available datasets. Their method achieves a cumulative match score of 90.5% for 2% of the data retrieved on their private dataset, including 210,000 impressions from Chinese crime scenes and searching through 210,000 images only takes about 30s.

Other works on real impressions include Tang et al. [TSKC11] and Kortylewski [KAV15, KV16], which model the impressions using primitive patterns. For example, Kortylewski et al. [KV16] represent each footwear impression using a hierarchical composition of active basis models in a bottom-up manner with Gabor wavelets as basis filters. Their goal is to learn a representation from reference prints that can then be used to find similar patterns in crime scene impressions. Figure 2.5 illustrates their approach in an example. The reference impression (b) is first used to learn the composition of active basis models (c). The model is spatially transformed during inference to find the optimal spatial configuration for a probe image (a) depicting a crime scene impression. To deal with partial impressions, they utilize a background model as missing parts will otherwise decrease the matching probability in these locations, similarly to cross-correlation approaches. On their publicly available dataset of 300 footwear impressions of real criminal cases and 1,175 reference impressions, Kortylewski et al. [KV16] achieve a cumulative match score of 55% for 10% of the database. However, since the optimal spatial configuration, i.e., each translation and rotation, has to be found for each probe image, the processing time is increased by multiples if the images are not properly aligned.

For other approaches published before 2017, Rida et al. [RBCP19] provide an extensive survey. More recently, deep-learning-based approaches [ZFDC17, KSRF17, KSRF19]

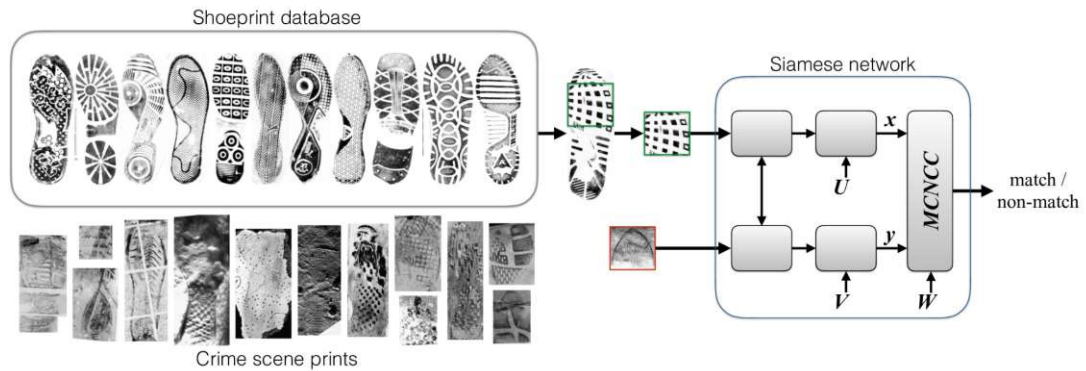


Figure 2.6: Siamese network with a Multi-Channel Normalized Cross Correlation (MCNCC) for footwear impression similarity by Kong et al. [KSRF19].

have been proposed for comparing footwear impressions. Zhang et al. [ZFDC17] use the features of the last layer before the Softmax of a VGG16 network and define similarity using a correlation coefficient. They propose using random pixel removal and simulated Gaussian noise to enlarge the number of training samples in a data augmentation step. Their VGG16 is pre-trained on ImageNet and fine-tuned using a classification loss on 88 high-quality impressions from the dataset published by Richetelli et al. [RLL<sup>+</sup>17]. They demonstrate using a closed-set-evaluations, see Chapter 5, on dust and blood impression of 18 pairs of shoes that their deep-learning-based method outperforms all others published in [RLL<sup>+</sup>17]. For example, compared to SIFT and RANSAC, the approach achieves a performance of 56.1% vs. 6.1% on the Dust test set. Unfortunately, they did not evaluate the FID-300 dataset [KV16]. Also, no assessment can be made of their performance in an open-set-evaluation where the shoe models in the test set differ from the training set. Furthermore, the number of samples of 18 pairs of shoes used for testing is limited and, as such, may not accurately represent performance on real-world impressions. However, since they represent each impression as a vector, an efficient search for similar impressions should be possible.

The currently best-performing approach for comparing realistic footwear impressions on the FID-300 dataset published by Kong et al. [KSRF19] is based on a Normalized Cross-Correlation utilizing features learned by a CNN. As shown in Figure 2.6 they use a siamese network with a paired regression loss which includes a Normalize Cross-Correlation extended for multiple channels. In order to improve cross-domain matching performance, i.e., crime scene impression vs. reference impressions, they use separate linear projections  $V$  and  $U$  of the feature vectors, which are learned jointly with a per-channel importance weighting  $W$ . Similar to other siamese networks, the weights of the CNN layers are shared. Using a fine-tuned ResNet50, they outperform Kortylewski et al. [KV16] significantly with a CMS uplift of more than 20% for 10% of the FID-300 images retrieved. Their approach allows matching partial impressions, which they tested using randomly sampled  $97 \times 97$  sub-windows. Similar to the approach proposed by



Techniques	Accuracy	Database Size	Studied Distortions
• (Bouridane <i>et al.</i> , 2000) [22]	88.00% @1	145	rotation & translation
• (De Chazal <i>et al.</i> , 2005) [23]	87.00% @5%	475	rotation & translation
• (Zhang and Allinson, 2005) [40]	97.70% @4%	512	rotation, noise, scale & translation
• (Pavlou and Allinson, 2006) [41]	85.00% @1	368	rotation & translation
• (Gueham <i>et al.</i> , 2007) [26]	100.00% @1	100	partial & noise
• (Gueham <i>et al.</i> , 2008a) [27]	95.68% @1	100	rotation, noise & occlusion
• (AlGarni and Hamiane, 2008) [29]	99.40% @1	500	rotation & noise
• (Gueham <i>et al.</i> , 2008b) [28]	99.00% @10	500	rotation, scale, noise & occlusion
• (Pavlou and Allinson, 2009) [45]	87.00% @1	374	-
• (Dardi <i>et al.</i> , 2009a) [47]	49.00% @1	87	noise
• (Nibouche <i>et al.</i> , 2009) [46]	90.00% @1	300	rotation, noise & occlusion
• (Patil and Kulkarni, 2009) [31]	91.00% @1	1400	rotation, noise & occlusion
• (Pei <i>et al.</i> , 2009) [32]	61.70% @5	6000	noise & occlusion
• (Dardi <i>et al.</i> , 2009c) [48]	73.00% @10	87	rotation, scale & translation
• (Tang <i>et al.</i> , 2010b) [50]	71.00% @1%	2660	rotation, scale, translation & occlusion
• (Tang <i>et al.</i> , 2012) [55]	70.00% @1%	2660	rotation, scale, translation & noise
• (Wang <i>et al.</i> , 2014) [57]	90.87% @2%	210 000	rotation, translation & scale
• (Kortylewski <i>et al.</i> , 2014) [58]	27.10% @1%	1175	translation & noise
• (Almaadeed <i>et al.</i> , 2015) [59]	99.33% @1	300	rotation, scale, noise & occlusion
• (Kortylewski and Vetter, 2016) [37]	71.00% @20%	1175	-
• (Alizadeh and Kose, 2017) [60]	99.47% @1	190	noise, rotation & occlusion

Figure 2.7: Comparison of the published results achieved by automatic footwear impression identification methods in a survey by Rida *et al.* [RBCP19].

Kortylewski *et al.* [KV16], a dense search over all translations and rotations is required in case the footwear impressions are not pre-aligned. The computational effort increase even further if other transformations like perspective transformations are considered. The query impression needs to be compared with each sample in the reference database using dense template matching. Therefore, their proposed methodology is not applicable for the fast retrieval of footwear impressions.

## Datasets

Since most publications do not use publicly available datasets for evaluation, a quantitative comparison between all published methods in this field is impossible. When looking at the survey by Rida *et al.* [RBCP19], this problem is made abundantly clear. In this survey, they compared 21 approaches using published results. The list of 21 published results shown in Figure 2.7 indicates that almost all publications use their private datasets for evaluation instead of publicly available ones. Moreover, even in case they use publicly available datasets for evaluation, like Kong *et al.* [KSRF19], they use private datasets for training. On the one hand, this makes it impossible to reproduce the results and, on the other hand, hard to compare the actual performance of these approaches without an error-prone re-implementation.

In 2014 Kortylewski *et al.* [KAV15] attempted to solve this problem by publishing the first, to the best of my knowledge, publicly available dataset of footwear impressions along with their approach based on using primitive patterns to describe the footwear impressions. Their dataset FID-300 includes 300 impressions from real criminal cases

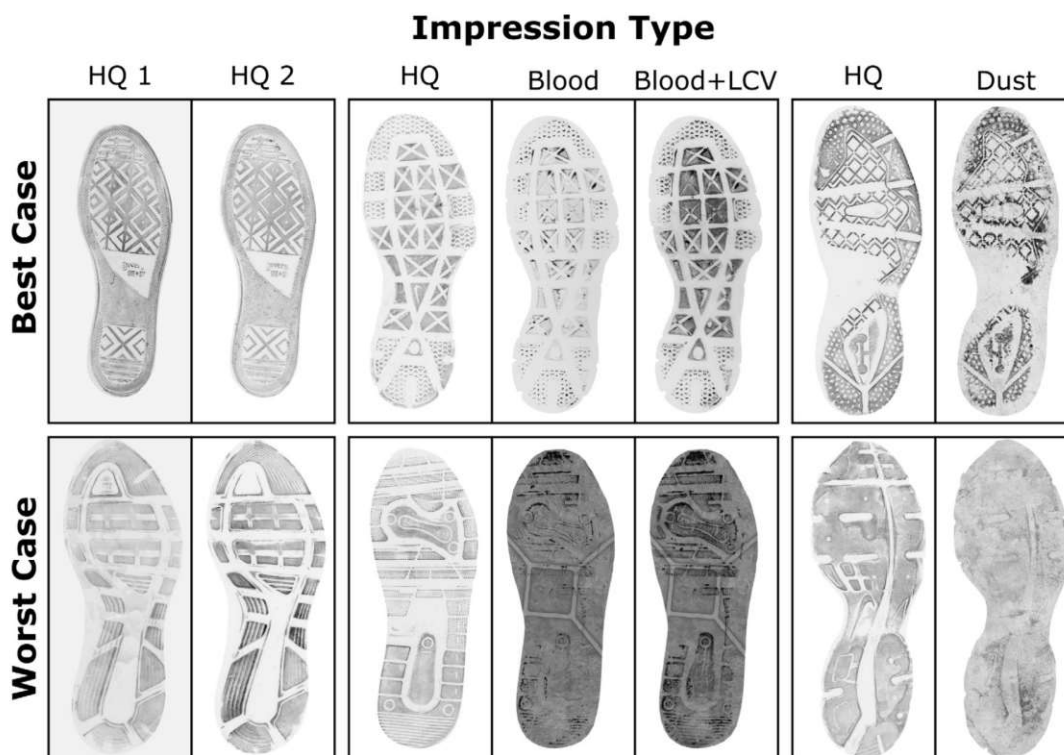


Figure 2.8: Examples of the footwear impression dataset published by Richetelli et al. [RLL<sup>+</sup>17].

and 1,175 reference impressions. Figure 2.5 shows an exemplary reference impression (a) with a matching crime scene impression (b). Additionally, to the cropped impressions, as shown in Figure 2.5, the original crime scene impressions that include rulers for scale are also provided.

Later in 2017, Richetelli et al. [RLL<sup>+</sup>17] used 18 pairs of shoes to create crime-scene-like impressions on different substrates using dust and human blood. However, their dataset only contains 180 impressions in total from 100 different shoes. The impressions were extracted from the background and registered in all the images. Figure 2.8 illustrates the different types of impressions included in the dataset.

Even though there are two footwear impression datasets publicly available, these datasets have some apparent limitations. Both datasets are not well suited for training deep learning algorithms since they provide only a few different classes (shoes) or only a few samples per class. Richetelli's dataset can not be used to train deep networks from scratch due to the limited size of the dataset, and the FID-300 was not designed to capture the variations among impressions created by the same shoe or shoe model. Nevertheless, in contrast to Richetelli's dataset, the crime scene impressions in the FID-300 cover a broad spectrum of impressions and include, for example, blurred impressions, overlapping

impressions, and even an image of a footwear impression made in the snow. However, the FID-300 ground truth is only based on the shoe model and not on the shoe; therefore, this dataset cannot be used to identify characteristics unique to a specific shoe. This problem is emphasized because the images are supplied with less than 1-megapixel resolution, making comparing individual characteristics not visible at that resolution impossible.

#### 2.1.4 Discussion

The previous section presented approaches for automatically comparing images of toolmarks, footwear impressions, and handwritings. For toolmarks, the presented approaches operate on 1D profiles of striated toolmarks and therefore commonly utilize techniques like cross-correlation for computing similarities. According to Baiker et al. [BHK<sup>+</sup>20] there is no prior work on automatically comparing toolmarks impressions, which explains the lack of approaches focused on two-dimensional local characteristics in this field. In contrast, for footwear impressions and handwritings, many of the presented approaches utilize either learned or handcrafted descriptors for describing local characteristics embedded into a global context. The presented approaches for writer retrieval commonly employ encodings like BOW, VLAD, or Fischer Vector for describing the distribution of local characteristics in handwritten pages. In contrast, methods shown for the automatic comparison of footwear impressions utilize registration approaches, like RANSAC, or template matching, like normalized cross-correlation, to find matching areas in the images. Since the distribution of local characteristics is important for writer retrieval and the positional relationship between the local characteristics for toolmarks and footwear impressions, this difference in describing the global context is intuitively explained.

For both writer retrieval and footwear impression comparison, learning-based approaches presented outperform other methods in this field. However, the available public footwear impression datasets either contain a limited variety for each class or a limited number of classes and are therefore not a great fit for training learning-based approaches. Toolmark datasets similarly only contain marks from 10 to 50 different tools. Contrary to that, various datasets with hundreds of writers and thousands of pages are publicly available for writer retrieval, and end-to-end based approaches have already been successfully trained on these datasets.

## 2.2 Metric Learning

Measuring the similarity of images is essential for many computer vision tasks like image clustering, face detection, and image retrieval [RMS<sup>+</sup>20]. One way to describe the similarity between images is to extract local features with a descriptor like SIFT and then model the similarity using these features [WSL<sup>+</sup>14]. Even though such local descriptors have worked well for tasks like writer retrieval, as shown by Fiel and Sablatnig [FS12], neural networks have shown to learn superior feature representations [SHSP17]. An extensive comparison of traditional descriptors with learned descriptors is provided by Ma et al. [MJF<sup>+</sup>21].

One method to train such models is to use a classification dataset, like ImageNet [RDS<sup>+</sup>15], train the model on this task, and then use the extracted features of a chosen network layer that provides the right level of abstraction as feature descriptors. Fischer et al. [FDB14] show that this approach can outperform SIFT for the matching and retrieval of images. Other approaches directly using the relationship between images to train neural networks will be the focus of this section. For example, this can be achieved by directly learning a similarity measure from pairs of samples similar to regression or binary classification with the classes “matching” and “non-matching.” During inference, both images are provided to the network to predict similarity scores.

Such an approach, with a siamese architecture with two network branches sharing weights, was first proposed for online signature verification by Bromley et al. [BGL<sup>+</sup>94] and independently for fingerprint verification by Baldi and Chauvin [BC93]. Baldi and Chauvin [BC93] use preprocessed images of fingerprints as input. Their network architecture similarly consists of two sub-networks, but they already use convolutional layers and a fully-connected decision network. More recently, this approach has been proposed for stereo matching patches using a siamese two-channel network [ZL15], shown in Figure 2.9. The weights of the layers L1–L3 are shared, and the final layer L8 projects the output to two real numbers that are subsequently fed through a softmax function, producing a distribution over the two classes “matching” and ‘non-matching.’ In these siamese architectures, the lower layers with shared weights can be seen as feature descriptor modules and the upper layers as similarity modules.

However, different forms of such architectures exist. Zagoruzkoa and Komadaski [ZK15] compare three such architectures, which are shown on the left and in the middle in Figure 2.10. In the 2-ch network, two patches are simply fed into the network, and the decision layer is trained to predict the “matching” and “non-matching” classes. These networks do not have a separable descriptor. Therefore, the patches have to be forwarded through the whole network during inference to predict a similarity score. In contrast to that, siamese and pseudo-siamese architectures depicted the middle in Figure 2.10 provide a separable descriptor and a decision network. During inference, the descriptors of all images can be computed independently and only need to be computed once. The similarity score is then predicted for each pair of images by the decision network using these descriptors. In contrast to the siamese architecture, the weights are not shared in the pseudo-siamese architecture, thus providing more flexibility [ZK15]. Zagoruzkoa and Komadaski [ZK15] further extend the concept by proposing a Central-surround two-stream network, shown in Figure 2.10 on the right, that utilizes a branch that receives a high-resolution center crop and a branch that handles a downscaled version of the whole image patch.

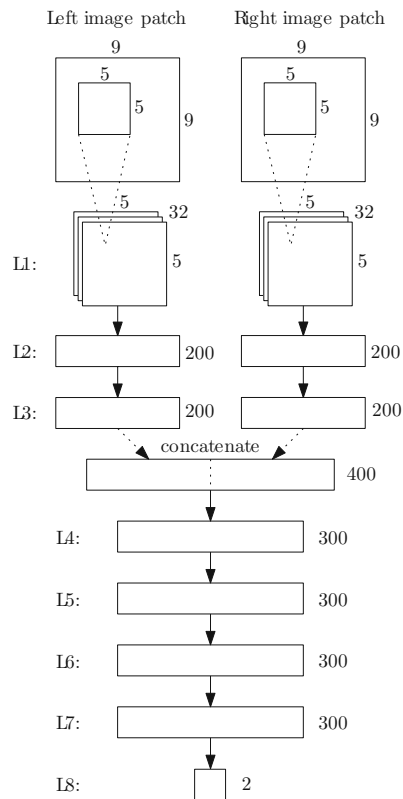


Figure 2.9: Two-channel network architecture proposed for stereo matching utilizing shared layers (L1–L3) [ZL15].

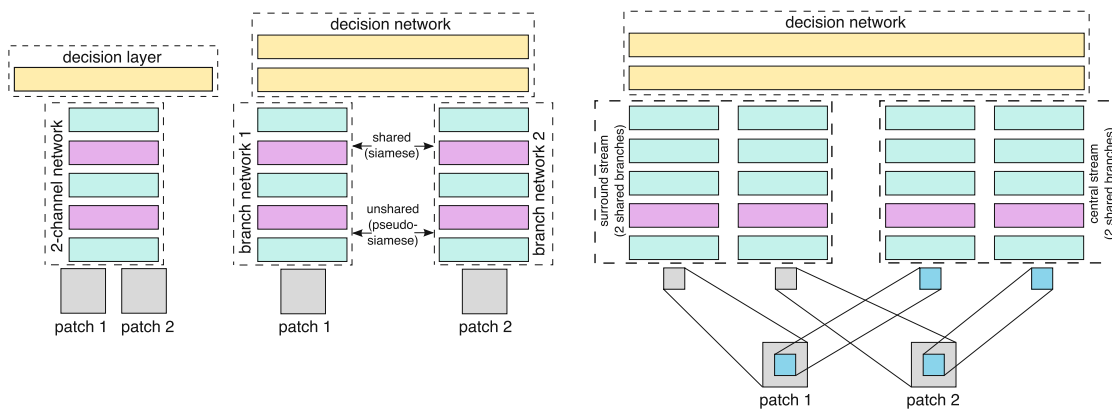


Figure 2.10: Different network architectures for learning image similarity. Two-channel without shared weights (left), siamese and pseudo-siamese (center), and Central-surround two-stream network (right) [ZK15].

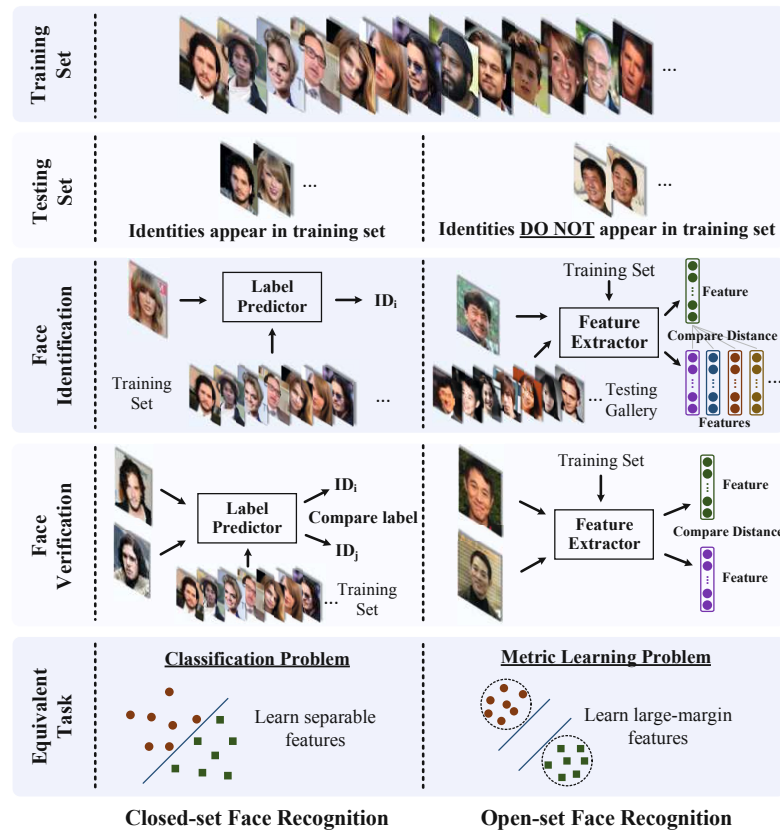


Figure 2.11: Metric learning vs. classification problem definition comparison on the example of face verification and identification [LWY<sup>+</sup>17].

In contrast to approaches that compute similarity for a pair of images directly, for so-called metric learning approaches, visual image similarity requires learning an embedding space that represents images and their similarity using a defined distance metric [RMS<sup>+</sup>20]. Methods that use deep neural networks to learn such a similarity preserving embedding are often referred to as Deep Metric Learning [RMS<sup>+</sup>20]. These metric learning approaches with CNNs will be the focus of this section. For applications like (fine-grained) image retrieval, zero-shot classification, face recognition, and clustering, learning such embeddings is increasingly important [RMS<sup>+</sup>20, ZW19]. Since similarities in the embedding can be efficiently computed using a defined distance metric like  $L_1$  norm,  $L_2$  norm, cross product, or similar, these methods are especially useful for image retrieval applications like image search, where the most similar image to a query image has to be retrieved since this speeds up the retrieval process significantly [MBL20].

Further, metric-learning-based approaches can be used when the semantic labels of the test samples are not the same as the semantic labels of the training samples, for instance, in open-set classification, where the classes in the test set are not the same as in the training set. For these applications, the embeddings must generalize well for unseen

samples during test time [ZW19]. An example of this is writer retrieval, for which, even though the classes are distinctly identifiable, i.e., the writers, a retrieval method should work not only for writers in the training set but also for unknown writers during testing. Similar applications are, for instance, face identification and verification, and person re-identification [MBL20]. In Figure 2.11 the difference between closed-set and open-set problem definitions is shown in the example of face recognition and a detailed description of different evaluations strategies is given in Section 5.

Embeddings can also be used if class labels are either unavailable at all or if it is time-consuming to acquire them. In this case, relationships between the sample can be used instead during training, e.g., pairs of matching and non-matching samples to learn the embedding [MBL20].

This section first describes ranking-based loss functions that operate on pairs, triplets, or larger sets of samples. The objective of these approaches is that the learned embedding should preserve these relative relationships between samples, e.g., similar samples should be closer in the embedding than dissimilar samples. For training, these methods only rely on these relative relationships. Semantic (class) labels can be used to create these pairs or triplets but are not required. In contrast, classification-based approaches that utilize semantic labels in training similarly to standard image classification methods are discussed afterward. Nevertheless, the line between these methods is blurry since some of these approaches allow a dynamic assignment of these labels without requiring pre-defined labels in the training samples.

### 2.2.1 Contrastive Loss

The siamese architecture with two network branches sharing the same weights was simultaneously proposed by Bromley et al. [BGL<sup>+</sup>94] and Baldi and Chauvin [BC93]. In contrast to other approaches that work on image data, online handwriting allows Bromley et al. to use stroke features like velocity, acceleration, and angle as input data for each point in time. They use two so-called Time Delay Networks in such a configuration to compare two signatures. Using backpropagation, they train the network to output a 38-dimensional feature vector for each input signature. In their approach, the angle between these feature vectors indicates if the signatures match (small angle) or if one signature is a forgery (large angle). This approach is significantly different from binary classification approaches described above, as the actual decision if two samples are similar is not computed using a part of the network but rather by a defined distance function, for which they use the cosine between the feature vectors. The angle  $\alpha$  between two vectors can be represented by the dot product of the (length normalized) vectors:

$$\cos(\alpha) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (2.1)$$

Therefore, in their approach the training is conducted by using  $\cos(\alpha) = 1.0$  and  $\cos(\alpha) = -0.9$  or  $\cos(\alpha) = -1.0$  as target values, respectively. Training is performed with backpropagation and MSE loss as proposed by LeCun et al. [LeC89].

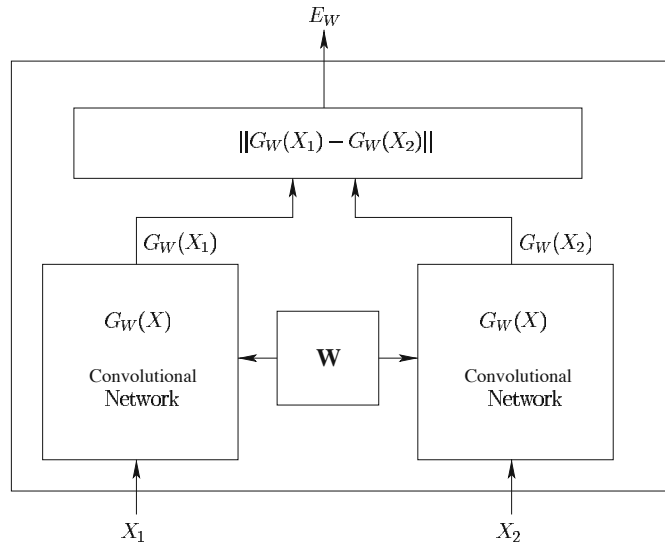


Figure 2.12: Siamese network architecture for face verification with an energy-based loss function [CHL05].

Chopra et al. [CHL05] build on this work and the work by Baldi and Chauvin [BC93] for the application of face verification. In their CNN siamese architecture, which is shown in Figure 2.12, two identical CNNs with shared weights are used to learn a low dimensional representation of input image pairs. In contrast to approaches described previously, each image is explicitly mapped into a feature space, where similarities between face images are computed using the  $L_1$  norm. The loss function is based on an energy-based approach, similarly to LeCun et al. [LH05]. The loss function's goal is to optimize this embedding so that the energy function between each sample in the training set and each similar sample is smaller than between the sample and each dissimilar sample by a margin  $m$ .

The same siamese network architecture is used by Hadsell et al. [HCL06]. They define their proposed contrastive function as follows:

$$L(W, Y, \vec{X}_1, \vec{X}_2) = (1 - Y)L_S(D_W) + (Y)L_D(D_W) \quad (2.2)$$

with a partial loss term  $L_S$  for similar pairs and a contrastive loss term  $L_D$  for dissimilar pairs. The exact loss function used by Hadsell et al. is then defined as:

$$L(W, Y, \vec{X}_1, \vec{X}_2) = (1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\max(0, m - D_W)^2 \quad (2.3)$$

with the Euclidean distance between the two feature vectors  $G(\vec{X}_1)$  and  $G(\vec{X}_2)$  as  $D_W$ . The parametric function  $G_W$  maps the input vectors  $\vec{X}_1$  and  $\vec{X}_2$  into the embedding space. Each input pair  $\vec{X}_1, \vec{X}_2$  is assigned a binary label  $Y = 0$ , i.e. similar, and  $Y = 1$ , i.e. dissimilar. The contrastive term  $L_D$  makes sure that the solution does not collapse to zero. Only negative pairs in the radius defined by the margin  $m$  are considered for the contrastive term in the loss function. In Figure 2.13 this is visualized using the



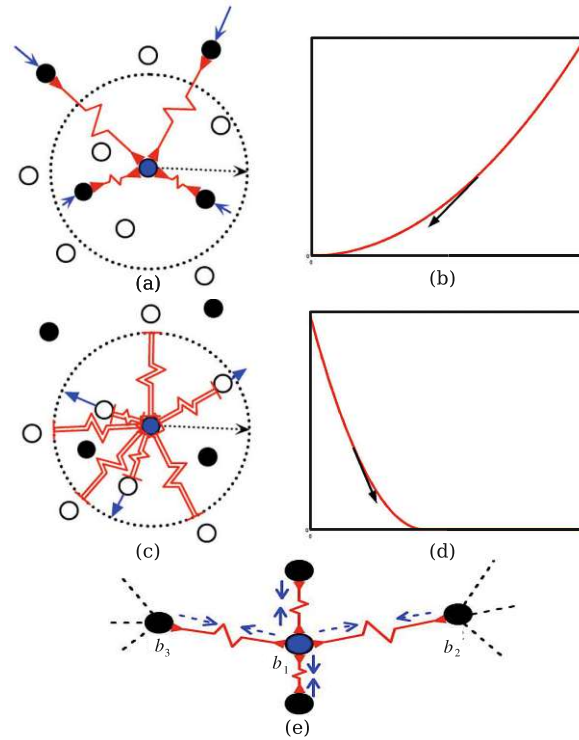


Figure 2.13: Spring model analogy for the contrastive loss [HCL06]. Similar samples (black) are pulled to the blue anchor (a), and dissimilar samples (white) are pushed to the margin (c). The corresponding loss functions are shown in (b) and (d), respectively. In (e), the equilibrium is shown after minimizing the global loss function.

spring model analogy by Hadsell et al. [HCL06]. In (a), similar samples (black) are pulled to the anchor with the red springs attached to each sample. The loss function for this process is shown in (b), which converges to 0. For dissimilar samples, they use *m-repulsive-only* springs, as shown in (c), which push dissimilar samples away until the margin  $m$  is reached. Therefore, the loss function in (d) converges to the margin  $m$ . By minimizing the global loss function, an equilibrium is reached, as shown in (e).

Commonly the contrastive loss is defined as [MBL20]:

$$L_{contrastive} = [d_p - m_{pos}]_+ + [m_{neg} - d_n]_+ \quad (2.4)$$

with the distances between similar and dissimilar pairs  $d_p$  and  $d_n$ , respectively, and the corresponding margins  $m_{pos}$  and  $m_{neg}$ . The hinge function is denoted with  $[\cdot]_+$ . As shown by Hadsell et al. [HCL06], the margin  $m_{pos}$  can be omitted. A potential issue with this approach is that the same margins are applied to all pairs without considering variances in interclass dissimilarity [MBL20] since relative distances between classes are not directly taken into account [MTL<sup>+</sup>17].

### 2.2.2 Triplet Losses

Instead of just computing the loss using pairs of similar and dissimilar samples, the triplet loss function attempts to include interclass variance into the loss function by the distance  $d_{ap}$  between an anchor and a positive sample smaller than the distance between the anchor and a negative sample  $d_{an}$  by a defined margin  $m$  [MBL20]:

$$L_{triplet} = [d_{ap} - d_{an} + m]_+ \quad (2.5)$$

Schroff et al. [SKP15] propose using the squared Euclidean distance as a distance metric, but also other metrics like the  $L_1$  norm [MTL<sup>+</sup>17] or the cosine [LWY<sup>+</sup>17] can be applied.

In contrast to this, Hoffer et al. [HA15] use a Softmax ratio of the two distances  $d_{ap}$  and  $d_{an}$  to create a ratio measure:

$$\ell(T) = \left( \frac{e^{\Delta^+}}{e^{\Delta^+} + e^{\Delta^-}} \right)^2 + \left( \frac{e^{\Delta^-}}{e^{\Delta^+} + e^{\Delta^-}} - 1 \right)^2 \quad (2.6)$$

with the  $L_2$  distance between the anchor and the positive sample  $\Delta^+$ , and the  $L_2$  distance between the anchor and the negative sample  $\Delta^-$ .

In contrast to this, Balntas et al. [BJTM16] propose a SoftPN loss which not only takes one negative distance into account but instead uses all three distances between the samples in a triplet [BJTM16]:

$$\begin{aligned} \Delta^+ &= \|f(x_{p_1}) - f(x_{p_2})\|_2 \\ \Delta_1^- &= \|f(x_{p_1}) - f(x_n)\|_2 \\ \Delta_2^- &= \|f(x_{p_2}) - f(x_n)\|_2 \end{aligned} \quad (2.7)$$

with the triplet  $T = \{x_{p_1}, x_{p_2}, x_n\}$  and the embedding  $f(x)$ . Instead of forcing the distance  $\Delta^+$  just to be smaller than  $\Delta_1^-$ , it is forced to be smaller than  $\Delta^* = \min(\Delta_1^-, \Delta_2^-)$ . The difference is illustrated in Figure 2.14. The distance  $\Delta^+$  corresponds to  $d_{ap}$  in the traditional triplet loss formulation as shown in Equation 2.5 when the sample  $p_1$  is used as the anchor and, similarly,  $\Delta_1^-$  corresponds to  $d_{an}$ . The distance between the positive sample  $p_2$  and the negative sample  $n$  is however not considered in the traditional distance formulation.

The loss is then defined as [BJTM16]:

$$\ell(T) = \left( \frac{e^{\Delta^+}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 + \left( \frac{e^{\Delta^*}}{e^{\Delta^+} + e^{\Delta^*}} - 1 \right)^2 \quad (2.8)$$

which is implemented using a Softmax layer and the Mean Square Criterion. The selection of training samples is simplified by this approach, as soft negative mining is performed implicitly [BJTM16] as illustrated in Figure 2.14.

Balntas et al. [BJTM16] evaluated their approach using the Photo-Tour dataset, which includes more than 500k image patch pairs with a size of 32x32 pixels. These pairs were

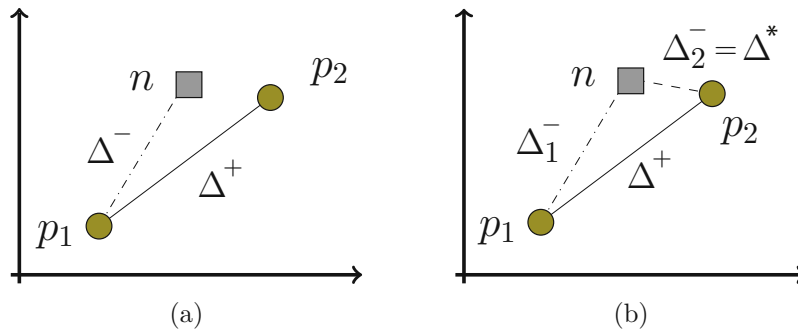


Figure 2.14: SoftMax Ratio (a) compared to SoftPN (b) [BJTM16].

Training		Notredame	Liberty	Notredame	Yosemite	Yosemite	Liberty	
Testing		Yosemite		Liberty		Notredame		
Descriptor	# features							mean
SIFT [15]	128	27.29		29.84		22.53		26.55
ConvexOpt [21]	≈ 80	10.08	11.63	11.42	14.58	7.22	6.17	10.28
DeepCompare <i>siam</i> [25]	256	13.21	14.89	8.77	13.48	8.38	6.01	10.07
<i>pseudo-siam</i> [25]	256	12.64	12.5	12.87	10.35	5.44	3.93	9.62
MatchNet [11]	512	11	13.58	8.84	13.02	7.7	4.75	9.82
<i>no bottleneck</i> [11]	4096	8.39	10.88	<b>6.90</b>	10.77	5.76	3.87	7.75
<b>PN-Net</b>	<b>128</b>	<b>7.74</b>	<b>9.55</b>	8.27	<b>9.76</b>	<b>4.45</b>	<b>3.81</b>	<b>7.26</b>
<b>PN-Net</b>	<b>256</b>	<b>7.21</b>	<b>8.99</b>	8.13	<b>9.65</b>	<b>4.23</b>	<b>3.71</b>	<b>6.98</b>

Table 2.1: Evaluation results of the PN-Net architecture on the Photo-Tour dataset compared with SIFT and learning based approaches [BJTM16].

extracted around specific feature points of landmarks like the Statue of Liberty. For sampling, they randomly choose a pair of patches from the same 3D point and randomly sample another patch from a different 3D point as the negative sample. As an evaluation metric, the False Positive Rate at 95% (FPR95) is used, which can be computed using the ROC curve, see Manning et al. [MRS08], by varying the threshold used to label pairs as matching depending on the computed distance. In Table 2.1 the reported results are shown in comparison to other learning-based approaches and SIFT, which represents traditional handcrafted feature extractors. The proposed method outperforms SIFT by a large margin even when using the same feature dimension of 128. The FPR95 is reduced from about 27% to 7% on average. The two-channel siamese approach described above DeepCompare [ZK15] is also outperformed.

The idea of incorporating the third distance in a triplet can also be applied to the triplet margin loss achieved in Equation 2.5 by swapping the anchor  $a$  and the positive sample  $p$  of the triplet if  $d(p, n) < d(a, n)$  [BLVM17]. In experiments performed by Balntas [BLVM17], such a triplet margin loss with anchor swap even outperforms the SoftPN approach.

### 2.2.3 Other Ranking-Based Losses

Other proposed losses go even further and try to utilize the relationships not only in a triplet but between all samples within a batch [ZW19]. These approaches identify informative samples to improve the convergence speed and accuracy of the trained models using batch sizes of 128 or more [ZW19]. Examples of these are the N-pair loss proposed by Sohn et al. [Soh16], the Lifted structure loss [OSXJS16], and Scaleable neighborhood analysis [WEY18], and L2Net [TFW17]. Some approaches, like Tuple Margin Loss [YT19] and FastAP [CHX<sup>+</sup>19], even utilize histograms to better learn the distribution of the batch in the embedding space. Similarly, Yuan et al. [YDT<sup>+</sup>19] propose the use of the ratio of signal variance to noise variance to improve the results. Wang et al. [WZW<sup>+</sup>17] investigate the influence of different similarities utilized by ranking-based losses and provide a Multi-Similarity loss that utilizes pair weighting and mining.

### 2.2.4 Classification Based Losses

One approach to compute image similarities using deep learning is to train a network on some classification problem, cut off the last fully connected layer and repurpose the features created by such a network to calculate image similarities. In this section, approaches that utilize classification-based losses and techniques are presented. In contrast to classical classification-based approaches, the semantic labels are only used during training with the goal of learning an embedding that represents the data faithfully and generalizes well. In their work, Movshovitz-Attias et al. [MTL<sup>+</sup>17] examine the connection between classification and ranking-based losses. They propose to learn a set of sample data points, i.e., *proxies*, as an approximation of all training samples. These proxies can either be assigned statically using class labels or dynamically using the proxy with the smallest distance to the sample. In contrast to the static assignment, where a sample's proxy is always the same, hence static, in the dynamic case, the sample's assigned proxy can change since the proxy and the points change in the backpropagation of the training. Dynamic proxies are needed if no semantic labels are available for the samples. For training, they use a Proxy-NCA loss which is based on [GHR04]:

$$L_{NCA}(x, y, Z) = -\log \left( \frac{\exp(-d(x, y))}{\sum_{z \in Z} \exp(-d(x, z))} \right) \quad (2.9)$$

In their formulation of the Proxy-NCA loss, they use the proxy of the input sample  $x$  as  $y$  and the set of all proxies as  $Z$ . In case of dynamic proxies, for the similar sample  $y$  the closest proxy. By minimizing this loss function, the probability that the sample  $x$  is closer to the proxy  $y$  than to any other proxy is maximized.

This is similar to the Softmax with negative log likelihood loss, which can be formulated as follows for one training sample [WWZ<sup>+</sup>18]:

$$L_s = -\log \frac{e^{f_{y_i}}}{\sum_{j=1}^C e^{f_j}} \quad (2.10)$$

with the input feature vector  $x_i$  and its semantic label  $y_i$  with the number of classes  $C$ . In this formulation the  $C$ -dimensional vector  $\mathbf{f}$  denotes the output of the last fully connected layer in the network before the Softmax function. Each element this vector can be defined as:

$$f_j = \mathbf{W}_j^T x_i + b_j \quad (2.11)$$

with the weight matrix of the fully connected layer  $\mathbf{W}$ ,  $j = 1 \dots C$  and the bias vector  $\mathbf{b}$ . This can be reformulated as function of the angle between then the weight vector for a class  $j$   $W_j$  and the input sample  $x_i$ :

$$f_j = \mathbf{W}_j^T x_i = \|W_j\| \|x_i\| \cos \theta_j \quad (2.12)$$

with  $x_i$  the  $i^{\text{th}}$  dimension of the one-hot encoded output vector  $x$  with the true class for the sample  $i$  and the number of classes  $n$ .

Movshovitz-Attias [MTL<sup>+</sup>17] argue that their loss offers an explanation for the effectiveness of using the output of the last layer before the fully connected layer of a cross-entropy trained classification network as an embedding. However, in their framework, the proxies are only used as intermediate representations of the whole sample space and discarded after training.

In contrast to other approaches that take advantage of bigger batch sizes, e.g., 128, to find informative triplets, their proposed method also works for smaller batch sizes. For example, in their experiments, they just use a batch size of 32. The proxy-based approach also significantly simplifies sampling in case semantic labels can be used since only the anchor has to be sampled directly. Since all proxies have to be kept in memory, for applications with a large set of semantic labels, they propose to assign these labels randomly to a smaller set of proxies. Their results show a performance drop from 66% to 59% (Recall@1 on Cars196 dataset) when using about half the number of proxies than labels. However, this is still significantly less than using a one-to-one mapping of semantic labels and proxies where a Recall@1 of 73% is achieved on the same dataset.

The biggest advantage of their method is that in their comparison, it converges about three times faster than methods that depend on finding informative triplet while still achieving a performance uplift of more than 20% (again Recall@1 on the Cars196 dataset) compared to, for instance, the FaceNet approach by Schroff et al. [SKP15] who use triplets with a semihard mining strategy. Interestingly, using proxies with such a triplet loss instead of the NCA loss also leads to a slight improvement of 3% (again Recall@1 on the Cars196 dataset), indicating that this might be a replacement for elaborate mining strategies. However, the uplift of using the NCA loss is more than 17% (again Recall@1 on the Cars196 dataset), which suggests that since the NCA loss function uses all proxies at once, the proxies can approximate the training space better.

A similar approach is proposed by Liu et al. [LWY<sup>+</sup>17] using a modified Softmax loss function that imposes an angular margin for the decision boundary. This angular margin is achieved by putting constraints on the weights of the last fully connected layer to map the samples onto a hypersphere. They argue that face images lie on a manifold,

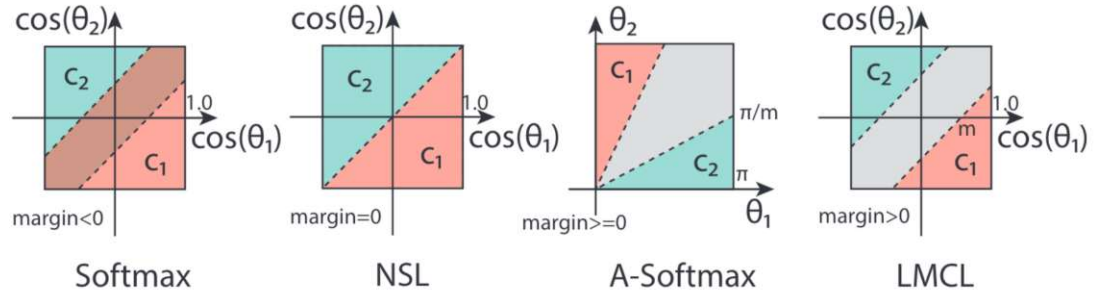


Figure 2.15: Decision margins compared for original Softmax, normalized Softmax, angular Softmax, LMCL in cosine space. Since the original Softmax is not defined in cosine space, the decision are overlaps [WWZ<sup>+</sup>18].

and thus restricting the embedding vectors to a hypersphere is reasonable. In their mathematical framework, they use the distance of a sample to the weight vectors  $W_i$  for each class, similarly to the proxies by [MTL<sup>+</sup>17]. However, their approach uses cosine as the distance metric instead of the  $L_1$  norm. Their strategy is similar to the one proposed by Zhai et al [ZW19], but instead of using a learned LayerNorm, the weight matrix  $\mathbf{W}$  is normalized, and the bias zeroed out. Furthermore, they define two separate decision boundaries, which are defined as follows for a two-class example [WWZ<sup>+</sup>18]:

$$C_1 : \cos(m\theta_1) \geq \cos(\theta_2) \quad (2.13)$$

$$C_2 : \cos(m\theta_2) \geq \cos(\theta_1) \quad (2.14)$$

with the angles  $\theta_1$  and  $\theta_2$  between the learned feature vector and weight vectors of the classes  $C_1$  and  $C_2$ , respectively.

Wang et al. [WWZ<sup>+</sup>18] extend this idea by changing the loss function. Instead of defining the margin  $m$  in the angular space, which makes it dependent on the angle, they define it in the cosine space. The decision boundary is defined as:

$$\cos(\theta_1) - m = \cos(\theta_2) \quad (2.15)$$

for a two-class example. Further, instead of just normalizing the class weights  $W_j$  they normalize also the input vectors  $x$ . Figure 2.15 compares the decision boundary for a two-class problem of the proposed approach Large Margin Cosine Loss (LMCL) to the A-Softmax proposed by Liu et al. [LWY<sup>+</sup>17] and normalized Softmax that normalizes the weight vectors of the class weights  $W_j$ .

Based on [MTL<sup>+</sup>17], Zhai et al. [ZW19] show that classification based approaches are applicable for image retrieval tasks. They use LayerNorm to Zero center the embeddings and a Normalized Softmax Loss which leads to a similar loss as proposed by [MTL<sup>+</sup>17] with the cosine function as the distance metric:

$$L_{norm} = -\log \left( \frac{\exp(x^T p_y / \sigma)}{\sum_{z \in Z} \exp(x^T p_z / \sigma)} \right) \quad (2.16)$$

with the weight  $p_i$  of a class  $i$  is viewed as a proxy and the proposed one-to-one assignment of proxies to class labels as proposed by Movshovitz-Attias et al. [MTL<sup>+</sup>17]. Since the cosine distance considers angles only, the bias term in the last linear layer is removed and an  $L_2$  normalization is applied to inputs and weights before the normalized Softmax. Layer normalization is applied to center the embeddings at zero. This simplifies their binarization of the embedding vectors which they propose to compress the representations. Further, they show empirically that this leads to a better initialization of the network.

As argued by Movshovitz-Attias et al. [MTL<sup>+</sup>17] the worst approximated samples within a class bound the loss. Therefore, Zhai and Wu [ZW19] propose sampling multiple samples per class for constructing the batches. Compared to the NCA loss proposed by Movshovitz-Attias et al. [MTL<sup>+</sup>17] they achieve a performance uplift of more than 8% (81% vs 73% Recall@1 on Cars196 dataset) with the same GoogleNet with Batch Normalization as the underlying network architecture. Although in their experiments they use an embedding dimension of 512 compared to 64 in the Proxy-NCA paper. Their results on the same dataset with a ResNet50 based architecture indicate that the embedding size impacts the performance as a drop from 512 dimensions to 128 leads to a performance decrease of 2-3% (84.2% to 81.6%). Even though they use different underlying model architectures to allow for a better comparison it is still hard to distinguish between performance differences due to network architecture, embedding size, normalization layers, hyperparameters, and actual impact of the proposed loss function. Their best performing network is able to significantly outperform the results achieved by Movshovitz-Attias et al. [MTL<sup>+</sup>17] with a Recall@1 of more than 89% on the Cars-196 dataset however this is achieved using a newer network architecture (ResNet50) and an embedding size of 2,048 which is 32 times what Movshovitz-Attias et al. [MTL<sup>+</sup>17] propose.

Zhai and Wu's [ZW19] proposed class balanced sampling also depends on bigger batch sizes. The batch size, however, is constrained by the memory footprint on the GPU [ZW19]. They utilize a batch size of 75 compared to 32 used by Movshovitz-Attias [MTL<sup>+</sup>17]. In their experiments they show, that a balanced sampling per class for each batch is beneficial for performance though since on the CUB-200-2011 dataset using just one sample per class in each batch leads to the worst performance of about 41% Recall@1 vs. the best performance of more than 61% by sampling 25 samples per class for 3 classes in each batch. On the CUB-200-2011 dataset they also show that the impact of Layer Normalization is more than 5%.

Similarly to [MTL<sup>+</sup>17], Zhai and Wu [ZW19] show that class subsampling during training is possible for applications with a vast amount of classes for which using all the classes is not feasible. An exciting aspect of their work is the proposed binarization of the embedding vector, which does not lead to a significant drop in performance (less than 1% Recall@1 on the Cars-196 from 88.7 to 89.3), suggesting that the actual magnitudes in each dimension of the embedding vector are not as important as if it is positive or not. This binarization compresses the 2,048 dimensional embedding vector to the same space as a 64-dimensional float vector. However, they did not investigate how subsampling or

quantization impacts the performance of other methods.

Other such classification based losses are, for instance, ArcFace [DGXZ19] who use an angular loss for face recognition, and Softtriplet loss [QSS<sup>+</sup>19] utilizing multiple proxies per class.

### 2.2.5 Discussion

The publications presented in the previous sections should give an overview of the field of metric learning and illustrate how metric learning approaches can utilize the relationship between samples to learn an efficient similarity measure for images. It established that metric learning is effective for learning a similarity measure for local features, and the PN-Net by Balntas et al. [BJTM16], for example, outperforms hand-crafted descriptors on the Photo-Tour dataset containing pairs of local features. Furthermore, other approaches, like [ZW19, MTL<sup>+</sup>17], demonstrate the applicability to end-to-end few-shot learning and fine-grained visual similarity of images containing bird species, cars, and online products. Similarly, Schroff et al. [SKP15] show the use for face recognition and clustering.

Recently three surveys have been published by Musgrave et al. [MBL20], Roth et al. [RMS<sup>+</sup>20], and Kaya et al. [KB19] that provide a more detailed comparison of approaches in this field. Unfortunately, in their in-depth evaluation, Musgrave et al. [MBL20] conclude that the publications have overstated the improvements achieved in the recent years. They used the same network architecture, embedding size, batch size, sampling, optimizer, and data augmentations to provide a fair evaluation. The source code for their benchmark is available<sup>2</sup> as well as their PyTorch-based metric learning Framework<sup>3</sup>.

Figure 2.16 shows the trend according to the results published over the years compared with the results achieved by the same methods in a fair evaluation. It demonstrates that even the contrastive loss published in 2006 is competitive with recent approaches. Consequently, the influence of the specific loss function used on the results achieved by metric learning approaches is marginal [MBL20].

## 2.3 Summary

First, this chapter presented related work for the automatic comparison of forensic images. A focus was put on domains discussed in this thesis, namely, toolmarks, handwritings, and footwear impressions. Nevertheless, a short introduction to finding similarities between other forensic samples like fingerprints was given as well. The related work on toolmarks presented includes automated comparison approaches and works concerned with the statistical foundation of forensic toolmark examination. In contrast to that, for handwritings and footwear impressions, this chapter focused on retrieval methods utilizing computer vision approaches. For all three forensic domains addressed, publicly

---

<sup>2</sup><https://github.com/KevinMusgrave/powerful-benchmark>

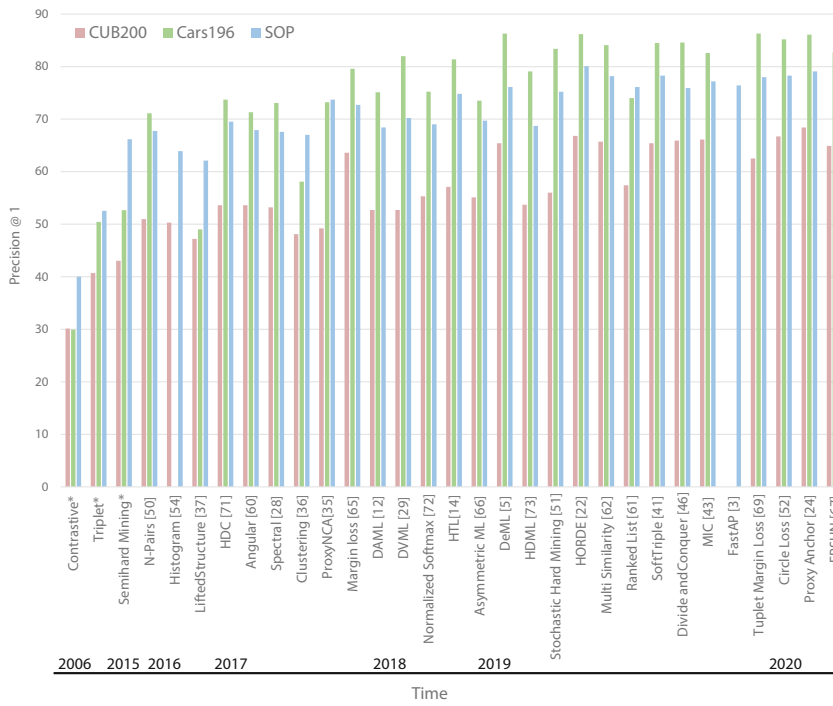
<sup>3</sup><https://github.com/KevinMusgrave/pytorch-metric-learning>



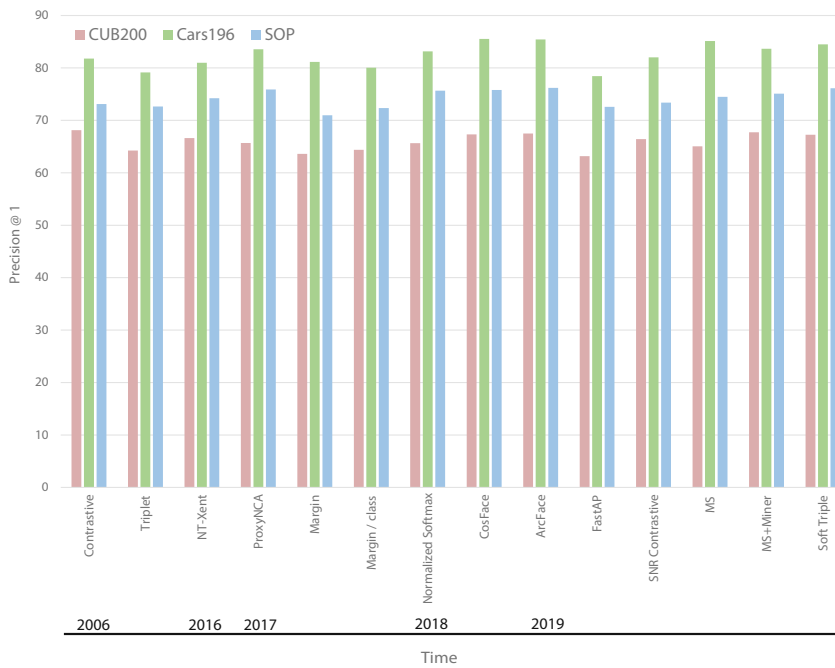
available datasets were presented. Finally, the presented approaches from all three forensic domains were discussed.

Secondly, this chapter presented computer vision methodologies for image comparison. It focused on metric learning approaches that use neural networks to learn an embedding space in which image similarity is represented by a defined distance metric. Nevertheless, approaches using traditional descriptors and other neural network architectures were also provided. For metric learning, approaches using different loss functions were presented to illustrate how these approaches utilize the relationships between samples to learn an embedding. This includes ranking-based loss functions based on the relationship between two, three, or more samples and classification-based losses that directly use the samples' labels. Finally, a discussion was provided that utilizes recent surveys to examine the performance of the presented approaches.

## 2. STATE OF THE ART



(a) The trend according to papers



(b) The trend according to reality

Figure 2.16: Published results over the years (a) vs. results achieved in a fair comparison (b) [MBL20].

# Datasets

Sample data with ground truth information is needed that represents the samples occurring in the ‘real world’ to predict the performance and objectively compare computer vision approaches. In addition to such an evaluation dataset, for applications that train models from sample data, similarly, a training dataset is required. The ground truth needed depends on the task at hand. For instance, for a classification task, labels indicating which object is present in the image samples are needed [RDS<sup>+</sup>15]. In case the object also needs to be localized, bounding box annotations can specify where the object is present in the image [RDS<sup>+</sup>15]. Even though in cases like the ImageNet dataset [DGXZ19] accumulating millions of images can be achieved with the help of internet search engines, creating ground truth information involves time-consuming manual labeling of images or correcting automatically created labels [DDS<sup>+</sup>09]. Even though a diverse representation with millions of samples is desirable to represent the ‘real world,’ this restricts the number of samples that can be feasible included in a dataset.

Datasets that do not depend on experts can outsource part of the manual labor. For instance, students wrote most of the handwritten pages in the CVL Database [KFDS13]. However, in many forensic domains, the dataset’s creation requires expert knowledge about the specific domain. For example, the NFI Toolmark Dataset by Baiker et al. [BKP<sup>+</sup>14] was created with an apparatus explicitly designed to create toolmarks with screwdrivers in a reproducible manner, as shown in Figure 3.1. Furthermore, they captured 2D images and 3D scans using forensic microscopes, which requires training for operation and therefore can not be outsourced easily. Thus, the dataset only contains toolmarks from 50 different screwdrivers, and other forensic datasets in this field similarly contain a small number of samples, e.g., 10 [BJJK10], 36 [PCDF<sup>+</sup>12], and 50 [SCE<sup>+</sup>15a]. Similarly, Richetelli et al. [RLL<sup>+</sup>17] only used 18 pairs of shoes to create footwear impressions. The other publicly available footwear impression dataset by Kortylewski et al. [KAV15] contains a lot more impressions, i.e., 300 from crime scenes and 1,175 reference impressions, and therefore captures the diversity of different shoe models better.

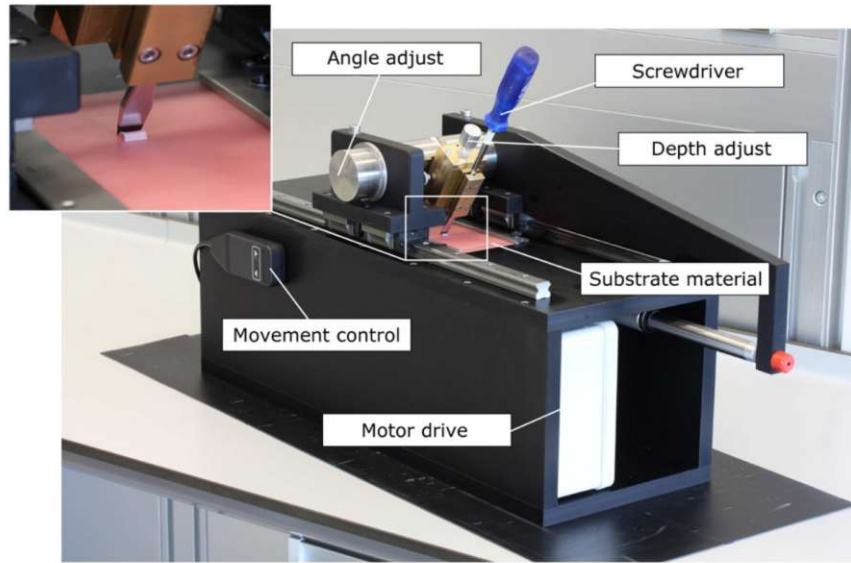


Figure 3.1: Apparatus designed by Baiker et al. [BKP<sup>+</sup>14] to create toolmarks of screwdrivers.

However, in contrast to Richetelli et al. [RLL<sup>+</sup>17], at most, two impressions of the same shoe model are provided, and hence, it does not represent the variations among different impressions created by the same shoe.

For writer retrieval, sufficiently diverse datasets already exist, like the aforementioned “CVL Database” [KFDS13] with five different handwritten pages by 284 writers. However, the available toolmark and footwear impression datasets are not adequate for the training and evaluation of the methodology proposed in Chapter 4. Therefore, this work presents two datasets containing more than 7,000 images of footwear impressions and toolmarks, created with the goal of designing an efficient workflow together with forensic experts to collect as many realistic samples as feasible. Both datasets were explicitly created to allow the training and evaluation of deep-learning-based methods. As such, they contain separate training and testing sets, multiple samples for each shoe and tool, diverse capturing conditions that mimic the workflow by the Austrian Police, and labels created together with forensic experts.

This chapter is divided as follows: first, the FORMS toolmark dataset is presented in Section 3.1, which includes the dataset’s creation, the annotation of the data, and a description of how the dataset is provided. Subsequently, the Impress dataset is presented similarly in Section 3.2.

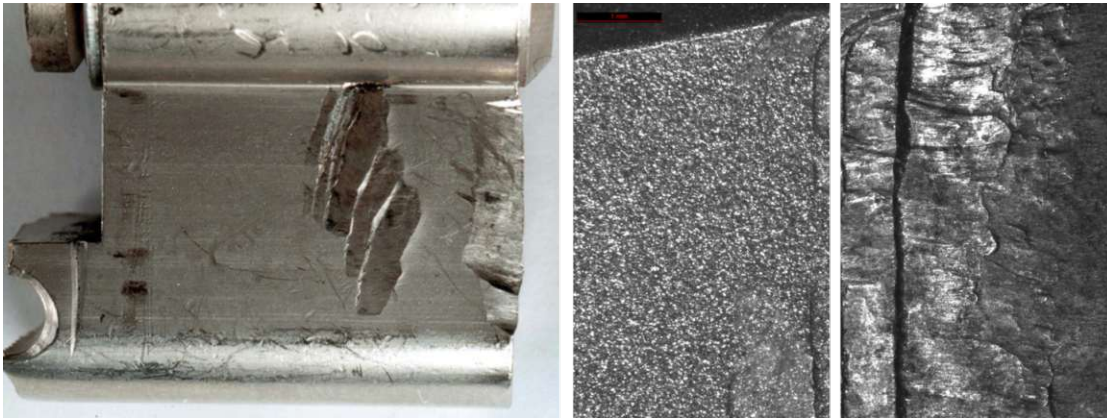


Figure 3.2: Image of a broken lock cylinder with toolmarks created by a locking-plier (left). Matching toolmarks on two lock cylinders photographed using a comparison microscope with a magnification factor of 20 (right).

### 3.1 FORMS Toolmarks

This section presents a new dataset with forensic images of toolmarks from real criminal cases to enable an evaluation of the real-world performance of toolmark comparison methods. This dataset consists of 3,046 toolmark images from 48 different crime series. It was created by photographing cylinder locks seized during criminal investigations of break-ins using a Leica comparison microscope.

This dataset aims to cover the current use case of the forensic experts of the Austrian Police: new lock cylinders are examined under a comparison microscope, and an overview image in  $10\times$  magnification, which contains the whole toolmark, is captured and archived. Similar toolmarks are then searched in a two-step process. First, the overview images digitally stored in the archive are compared manually. Secondly, the actual cylinders are retrieved and compared in  $20\times$  magnification under the comparison microscope if a potential match is found. Finally, an image of the aligned matching parts of the toolmarks is saved as evidence for court if the expert confirms the match.

Since the camera attached to the microscope examined for this work has a restricted resolution of 5MP, the striated patterns of the toolmarks are not always visible. Therefore, the focus of this dataset is put on matching impression marks left by the edge of the tool instead of striation patterns discussed in other publications mentioned in Section 2.1. Furthermore, as shown in Figure 3.2 on the right side, extracting the foreground (the toolmark) from the background (lock cylinder) is challenging due to the varying background structure and impression depth of the toolmark. Hence, the toolmarks impressions are annotated in the provided images. For this, a plugin for the image viewer *nomacs*<sup>1</sup> was developed, allowing the efficient annotation of matching

<sup>1</sup><https://nomacs.org/>

patches in the images to provide matching local image similarities. It allows the definition of polylines to describe the toolmark edges, and matching toolmarks can be fitted using translations and rotations. Matching toolmarks can either be found on the same lock cylinder or on different lock cylinders from the same linked case. Since all cylinder locks photographed originate from linked cases, it is guaranteed that multiple toolmarks exist in the dataset for each tool.

In order to allow an evaluation of the influence of different lighting conditions, all images were captured under 11 different lighting settings. Especially in combination with differences in lock materials and force applied, the appearance of the toolmarks varies significantly, as shown in Figure 3.2 on the right. Capturing the images under varying conditions enables training-based methodologies to learn robust models that work in real-world conditions and not just in a fixed laboratory setting.

Similar to the Photo-Tourism dataset [WB07], patches and matching and non-matching pairs are made available to allow a quantitative performance comparison of local image similarity-based methodologies. Additionally, the original images, manual annotations, and the annotation tool are provided. 197 lock cylinders from 48 linked cases were photographed on both sides. The resulting 3,046 images are divided into a training set and testing set by year, i.e., 2015 for training and 2016 for testing.

In this section, first, the dataset's creation and the ground truth annotation tool are described in detail. Subsequently, the published dataset with three different partitionings and the file format of the annotations are explained. An evaluation is performed later on in Section 5.3.

#### 3.1.1 Creation

This section first describes the image acquisition process for the dataset in detail. Since the dataset's creation is motivated by the needs of forensic experts, it is based on the current workflow at the Austrian Police. Secondly, the annotation of matching points in the toolmark images and the tool developed for this purpose are presented.

#### Image Acquisition

Lock cylinders seized by the Austrian Police during break-in investigations in Vienna in 2015 and 2016 were used to create the dataset. The comparison of toolmarks is conducted by the forensic experts using a Leica comparison microscope with lenses of varying magnification factors and an attached digital camera with a resolution of 5MP. Figure 3.3 depicts the comparison microscope used, and Figure 3.4 shows a closeup of the holding plate where the lock cylinders are placed. An adjustable ring light varies the lighting conditions with 11 different settings to enhance the contrast of specific toolmark characteristics. Additionally, a flexible spotlight can be utilized for this task. Figure 3.5 depicts the different settings of the ring light, and the influence on the visible toolmark patterns is illustrated using toolmark image crops.



Figure 3.3: Leica comparison microscope which was used for capturing the toolmark images.

New lock cylinders are first cataloged during a typical workflow using a  $10\times$  magnification lens to create an overview image containing the whole toolmark or toolmarks in case multiple marks are present. Since both jaws of the adjustable wrenches used in such break-ins have independent characteristics, both sides of the cylinders are captured. By consistently placing the cylinders upright with the broken (inner) part of the lock facing left, all toolmark images can be compared without rotating or flipping the images. Thus, this was done in the same way to create the dataset. Even though striated toolmarks are better visible under  $20\times$  magnification, it is not possible to capture the whole toolmarks with one image with the available setup since this requires stitching multiple images, which introduces artifacts at the borders. Furthermore, it complicates and lengthens the capturing process significantly and therefore contradicts the initial objective to alter the workflow of the forensic experts as little as possible.

Since one goal of this dataset is to investigate the influence of different lighting conditions and the robustness of a similarity measure to variations in lighting, each side of the cylinder was captured with all 11 available lighting settings. These settings, which are

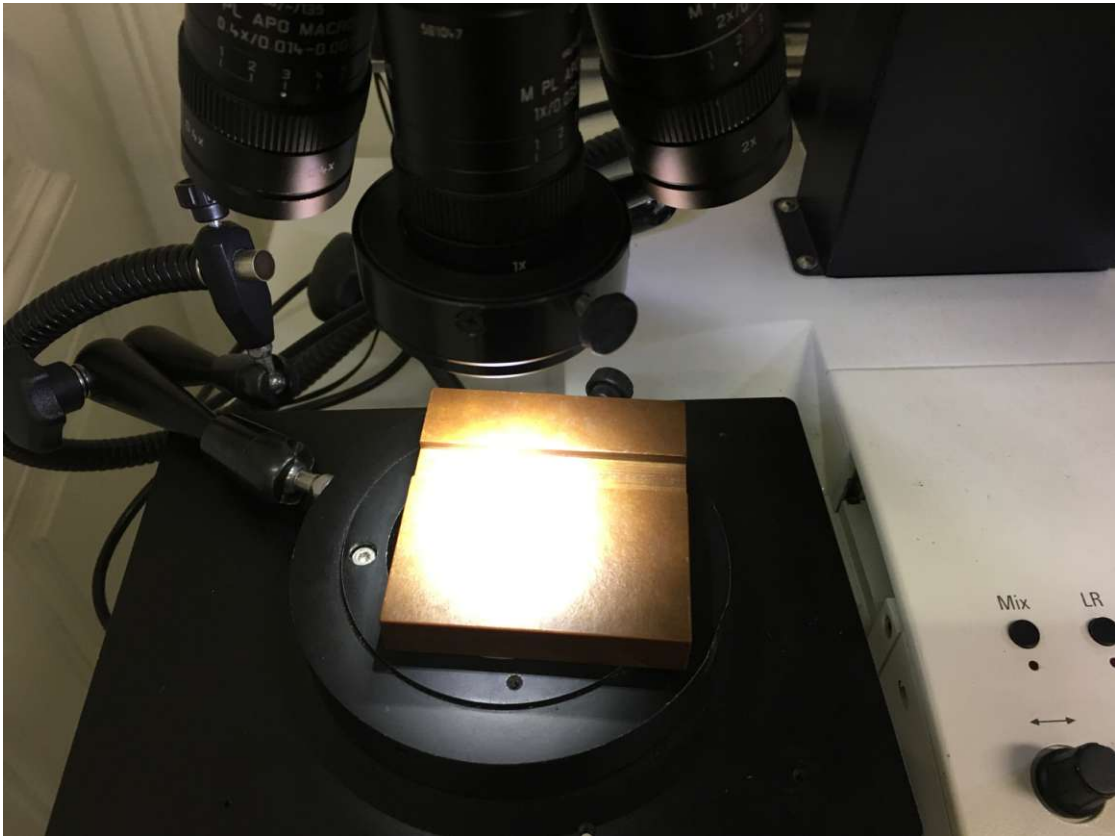


Figure 3.4: Closeup of the holding plate of the comparison microscope. The notch in the plate guarantees that the locks are inserted upright and the surface is flat.

divided into four different groups, are shown in Figure 3.5. In each group, different fractions of the ring light are lit up, and in each group (except *full* illumination), the direction of the light can be changed. This information is made available by coding the lighting condition into the image names, i.e. images with filenames ending with “01” belong to group *full*, “02”, “03”, “04”, “05” belong to group *half*, and so on. For images of cylinders from the year 2015, the file ending also indicates the exact lighting direction, e.g., “06” indicates the group *quarter* and direction from the top. Due to an issue with the light ring, the group can only be derived for the cylinder images from 2016. In Table 3.1 the number of tools, locks, and captured images in total are shown for each year.

As single toolmarks without matching counterparts cannot be used for evaluation and training, this dataset focuses on seized lock cylinders that have been previously identified as matching by forensic experts. This restriction limits the number of different tools in the dataset to the 48 tools used in linked criminal cases, i.e., crime series. In total, the dataset contains 3,046 images.



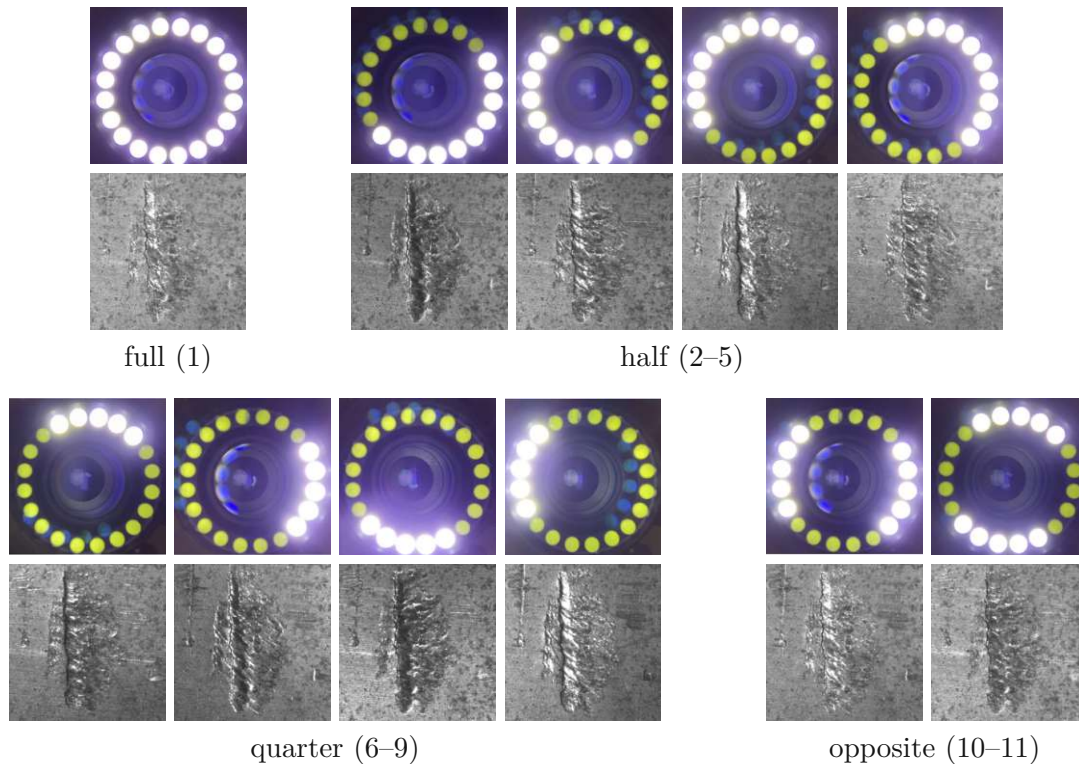


Figure 3.5: The 11 different lighting settings of the ring light used which are organized into 4 different groups. To illustrate the influence on the toolmark images, corresponding crops are shown below.

	Tools	Locks	Sides	Images
2015	25	115	230	1,782
2016	23	82	164	1,263
total	48	197	394	3,046

Table 3.1: Statistics of the captured toolmark images divided by year.

### Ground Truth Annotation

Even though the toolmark images acquired in the previous section are already annotated with the tool used, only 48 distinct tools with 96 distinct sides, i.e., jaws, are available. To provide a more fine-grained annotation of the images, matching local toolmark characteristics, i.e., matching patches, are desired. Manual annotation of matching patches in the toolmark images is not feasible since hundreds of patches per image have to be matched. Nevertheless, the toolmark images provide advantageous properties to simplify this task. Since the regions of interest lie on the edges of the toolmarks, a polygonal chain (polyline) can be used to describe the points on this edge parametrically. Further, the same polyline can be used to describe the edges in matching toolmark images,

albeit transformations are required to fit a polyline to a new image. The transformations necessary are given by the image acquisition process and the properties of the lock cylinders and jaws of the adjustable wrenches. First, the area of interest containing the toolmarks on a cylinder lock is approximately a flat surface, and the capturing angle is orthogonal to this surface. Therefore, no perspective transformations are necessary. Further, the distance of the camera to the surface is always the same since the lenses used have a fixed focus, and therefore focusing of the image is performed by moving the lock cylinder into focus. Thus, also scaling transformations do not have to be considered. By restricting the allowed transformations to translation and rotation, the polylines can be efficiently fitted to the edges of matching toolmarks.

The described approach was implemented as a plugin for the image viewer `nomacs`. Similar to `nomacs`, the so-called `PatchMatchingPlugin` is open source and available on GitHub<sup>2</sup>. The developed tool allows the user to draw polylines along the edges of the toolmarks. Further, polylines can be cloned and manually fitted to a matching toolmark in the same image using rotations and translations. The resulting polylines and their clones then define matching points on their line segments. Matching patches can be extracted by choosing one point on a polyline and using the transformation matrices to map this point to a clone of this polyline. The patches on a polyline and its clones can be displayed with varying patch sizes and distances between the locations on the polyline to assist the annotation process. For that annotation, matching toolmarks on two different cylinders and multiple distinct toolmarks on one cylinder are considered. Since the `PatchMatchingPlugin` only allows the annotation of a single image, first, for both sides of each cylinder, the user chooses the image in which the toolmark is best visible. After that, matching images are merged into one image, which is then used for the annotation process.

Figure 3.6 shows the annotation result in an example. At the top, the merged image is shown. In this case, three distinct toolmarks are visible in the image on the left side and two toolmarks on the right side. By drawing a polyline, a toolmark edge is first marked; in this figure, the red one. Then, for each distinct toolmark, a clone is created and manually fitted using translation and rotation. At the bottom, extracted patches along the polyline and corresponding points on the clones are shown to help the user adjust the fitting. Color coding is used to associate the clones with their respective patches. This association is done by coloring the control points of the polyline differently for each clone. The interpolated points between the control points are shown in gray. Finally, the toolmark's polyline and the transformation matrices for the clones are stored in a JSON (JavaScript Object Notation) file. Since the lock cylinders were not moved during the image acquisition, the same annotations can be used for the images of all lighting settings.

The annotation process was performed for all 96 merged images in the dataset, and a forensic expert verified the results. The example depicted in Figure 3.6 can be considered

---

<sup>2</sup><https://github.com/nomacs/nomacs-plugins>

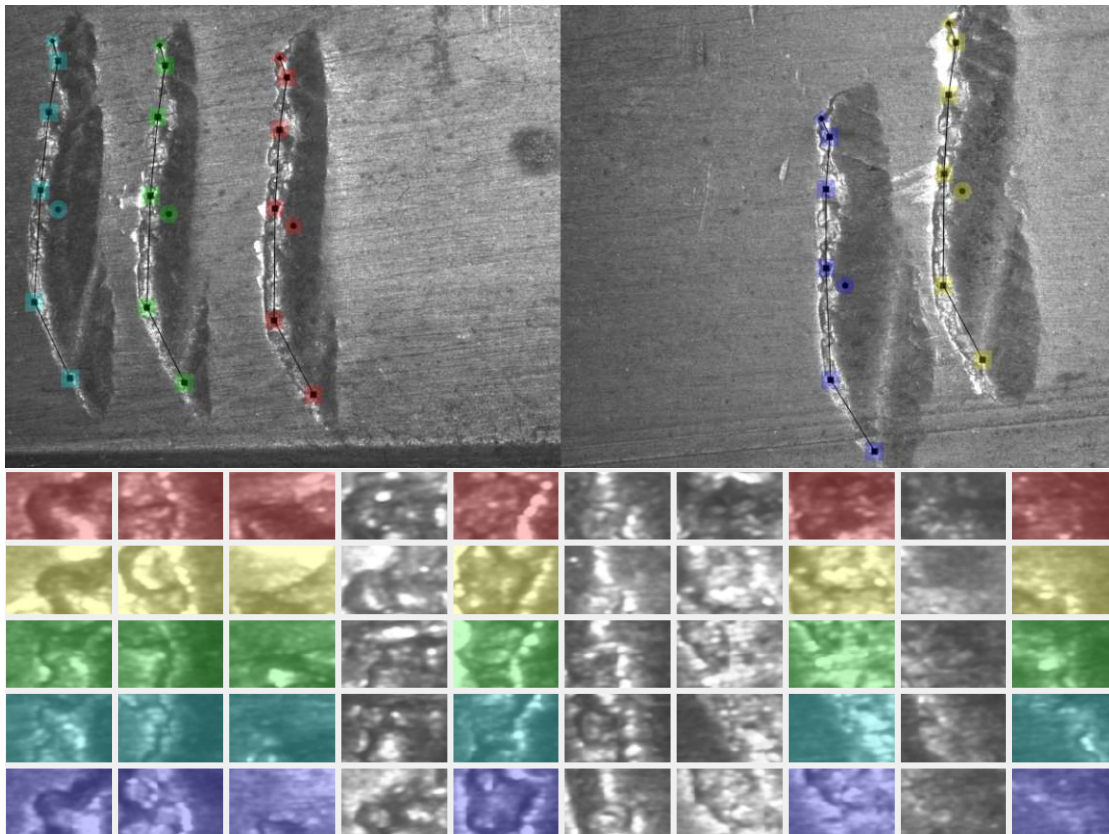


Figure 3.6: Annotation tool which assists the manual drawing, cloning and fitting of polylines. The merged images, which show multiple toolmarks made by the same tool, are displayed at the top. For each toolmark, a drawn polyline has been cloned and manually fitted to its edge using translations and rotations. Patches are extracted along these polylines to aid the user at precisely aligning the polylines. This is shown at the bottom. Control points on a polyline and corresponding patches have the same color.

a best-case annotation result. Depending on the hardness and shininess of the lock cylinder material and the force used by the intruder to create the toolmarks, the fitting of the different clones may be significantly worse. In particular, if a toolmark imprint is not deep enough, reference points which are crucial to aligning the clones, like the start and end of a toolmark or distinct patterns, may not be clearly visible. Further, in case overlapping toolmarks are present, finding a clear, consistent toolmark is challenging.

### 3.1.2 Dataset

In this section, the two ways the dataset<sup>3</sup> is provided are described. First, the 96 merged images for each jaw of a tool with their respective annotations and 11 lighting settings,

<sup>3</sup><https://cv1.tuwien.ac.at/cv1/forms-locks/>

i.e., 1,056 images and 1,056 JSON files. Secondly, to allow a comparative evaluation of different similarity measures without a separate patch extraction, image patches are provided with a list of 100,000 matching and non-matching pairs. For this, three distinct partitionings focusing on different challenges of the dataset are presented.

### Annotated Images

On average, 2.9 toolmark images were combined to create 96 merged images for each jaw of a tool. In Table 3.2 the detailed distribution is depicted. In most cases, only two toolmark images per side are available. However, in some cases, as many as ten different toolmark images of the same tool could be combined. In a few cases, no toolmarks were visible on one side of the lock cylinder, and therefore no image was captured.

	Matching Toolmark Images (per side)									
	1	2	3	4	5	6	7	8	9	10
2015	3	20	8	11	4	1	0	1	0	2
2016	8	22	8	4	5	2	0	0	0	0
total	11	42	16	15	9	3	0	1	0	2

Table 3.2: Distribution of matching toolmark images in the dataset.

An annotated image consists of one merged image with a minimum dimension of 2592x1944 pixels and a JSON file of the same name with the prefix “patches.json”. The images are concatenated by placing them side by side. Therefore, the width of the merged images is a variable multiple of 2,592 pixels. The filename contains the year (“15” or “16”), the side (“1” or “2”), and the lighting configuration (“01”-“11”, as depicted in Figure 3.5) for each tool. For example, the image showing toolmarks of the second side of the tool with index 1 in the year 2016 captured under light settings 3 (group *half*) is named “1\_16\_2\_03.png”. In Table 3.3 the number of polylines and clones in the dataset are shown. A merged image can contain multiple polylines to describe partial toolmarks and none if no toolmarks are visible.

	Merged Images	Polylines	Clones
2015	50	52	175
2016	46	44	115
total	96	96	190

Table 3.3: Number of polylines and clones in the dataset.

The annotated polylines and clones are stored in a JSON file. An example is shown in Listing 3.1. The top-level structure is an array containing one or multiple entries with polylines (called “polygon”) and clones. The polylines are defined by their control points with an array of 2-dimensional image coordinates. For each clone, a 3x3 transformation matrix is given, which allows mapping the points on the polylines to the actual image coordinates.

Listing 3.1: Exemplary JSON annotation with one polygon described by two control points and four clones of this polyline defined by their transformation matrices.

```
[{
  "polygon": {
    "points": [
      [3807.3409391715777,
       779.96621476630935],
      [3813.4372570232063,
       1334.7311392645361]
    ]
  },
  "clones": [{
    "transform": [
      [1,0,0],
      [0,1,0],
      [0,0,1]
    ]
  }, {
    "transform": [
      [1,0,0],
      [0,1,0],
      [2689.1673532866389,
       -103.6374034776909,1]
    ]
  }, {
    "transform": [
      [1,0,0],
      [0,1,0],
      [5011.6001511800332,
       -253.80000832021267,1]
    ]
  }, {
    "transform": [
      [1,0,0],
      [0,1,0],
      [-2618.6000989574227,
       145.80000477969634,1]
    ]
  }
]}]
```

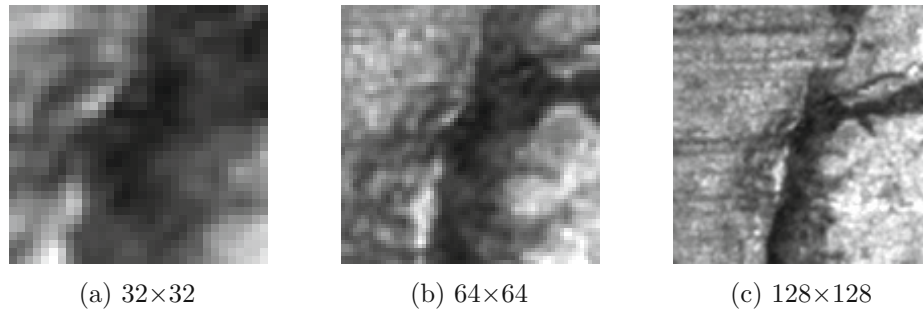


Figure 3.7: Three different patch sizes extracted at the same location

### Extracted Image Patches

In order to allow a comparative evaluation of methods for local toolmark similarities, a training set and testing set of extracted patches are provided. The sets are divided by year since the number of tools for 2015 and 2016 are approximately the same. However, as shown in Table 3.1 more images are available for the year 2015. Further, Table 3.2 indicates that the merged images for the year 2015 contain more images on average, which is crucial to providing matching patches for training. Therefore, 2015 is used as the training set, and patches extracted from 2016’s images are used for the testing set.

As described in Section 3.1.1, each location on a clone in the dataset is defined by the position on the polyline, i.e., the distance from the first control point calculated by following the line segments, and the transformation matrix for the clone.

The extraction of patches is performed as follows: for each clone in each merged image,  $64 \times 64$  patches are extracted along the polyline with a stepsize  $k = 64$  and  $k = 8$  for the testing set and training set, respectively. As shown in Figure 3.7, a patch size of  $64 \times 64$  shows enough details in the immediate area around the impression edge without too many details of the lock surface. Each patch in a dataset can therefore be uniquely identified by the merged image with the specific lighting configuration  $L$ , the polygon index  $I_p$ , the location on the polygon  $t_i$  indexed in steps of  $k$ , and the clone index  $I_c$ . The filenames of the patches are then composed by a counting number, index of the tool, year, side, “fg” for foreground,  $I_p$ ,  $t_i$ ,  $I_c$ , and  $L$ . For example, “025795\_30\_15\_2\_fg\_01\_0011\_02\_03.png” shows the patch #25795, extracted from the merged image “30\_15\_2\_03.png” on the 11th position on the first polygon, on clone 2 and with lighting configuration 3. Only patches on the polylines are extracted; patches in the background are ignored. Three distinct partitionings of the dataset FORMS-Locks, FORMS-Locks-RR, and FORMS-Lock-Lighting are proposed:

**FORMS-Locks** This partitioning focuses on finding matching patches without considering their orientations. Therefore, the orientation of each patch is fixed to the rotation of the transformation matrix of the corresponding clone. This way, matching patches are guaranteed to be oriented the same. In this partitioning, patches are defined as

“matching” if they are extracted from matching positions of clones in any lighting setting.

**FORMS-Locks-RR** This partitioning is similar to FORMS-Locks, yet, the robustness regarding variations in orientation is evaluated additionally. For this, patches are extracted with random orientations for the testing set. Ten patches with random orientations are extracted for the training set at each location. That increases the number of patches in the training set 10-fold compared to FORMS-Locks. “Matching” patches are equally defined as in FORMS-Locks.

**FORMS-Locks-Lighting-RR** This partitioning aims to isolate the influence of varying lighting conditions on the performance. Therefore, to remove errors introduced by the manual annotation of the matching toolmarks and variations due to cylinder lock materials and force applied, matching points on clones are ignored, and only patches from exactly the same image location but with different lighting settings are considered as “matching” patches. Similar to FORMS-Locks-RR, the patches are extracted with random orientation.

Table 3.4 shows the number of patches in each partitioning. The lists with 50,000 randomly sampled matching and non-matching pairs from the testing sets for evaluation are provided as CSV files which include two patch indices and a “0”/“1” indicator in each line for matching and non-matching pairings, respectively. Further, in addition to the  $64 \times 64$  sized patches, scaled-down  $32 \times 32$  patches are provided for all partitionings.

	#Patches	
	train	test
FORMS-Locks	41,030	25,014
FORMS-Locks-RR	410,300	25,014
FORMS-Locks-Lighting-RR	410,300	25,014

Table 3.4: Number of patches in the datasets.

### 3.1.3 Conclusion

This section presented a new toolmark dataset based on real break-ins investigated by the Austrian Police. Since no similar dataset exists yet, this contribution is crucial for developing methods for the automatic comparison of toolmark images. It is extensively described how the dataset was created and manually annotated. In addition to the 3,046 captured images and annotations describing matching points in these images, the annotation tool is also made publicly available. Further, three different partitionings, with more than 25,000 patches in the testing sets, are provided to allow quantitative comparisons.

## 3.2 Impress Footwear Impressions

Footwear impressions are commonly found at crime scenes and are thus a valuable source of evidence for criminal investigations. Forensic experts can show that a footwear impression was made by a specific shoe or the same suspect made impressions at different crime scenes by comparing individual characteristics of the impressions. However, this process is very time-consuming, and therefore, automated solutions are desired. However, testing and training such methods requires datasets that, on the one hand, reflect real data from criminal cases and, on the other hand, provide ground truth information. Hence, in this section, a new footwear impression dataset is presented. For this, an acquisition line was created, and footwear impressions of 300 different pairs of shoes were captured under varying conditions with the help of the Austrian Police. In this section, the creation of this dataset, and the dataset itself, are described in detail.

Section 2.1 shows that most publications on the automatic comparison of footwear impression do not use publicly available datasets for evaluation, and therefore a quantitative comparison between all published methods in this field is almost impossible. Even though there are two footwear impression datasets publicly available, these datasets have some apparent limitations. Both datasets are not well suited for training deep learning algorithms, which is unfortunate since, as mentioned above, those algorithms are the current state of the art in this field. Richetelli's dataset is too small to train deep neural networks from scratch, and the FID-300 was not designed to capture the variations among different impressions created by the same shoe. Additionally, the FID-300 ground truth is only based on the shoe model and not on the shoe, and therefore, this dataset cannot be utilized for comparing individual characteristics of impressions. This drawback is emphasized because the images are supplied in less than 1-megapixel resolution, making comparing individual characteristics impossible. Even though Richetelli's dataset provides high-resolution 600dpi scans, a total number of 18 pairs of shoes is not enough for most applications.

In order to mitigate these shortcomings, the dataset created presented in this section provides:

**Size:** 300 different pairs of shoes

**Variance:** multiple modalities (gelatin foil lifters, reference impressions, 3D molds, etc.) created on multiple surfaces (wood, paper, etc.)

**Image quality:** high quality scans of at least 600dpi or images taken using a DSLR

The dataset was created using an acquisition line where participants walk along a given path to produce predefined footwear impressions. This strategy provides an efficient way to create crime scene-like footwear impressions.

The following sections describe the acquisition process and the different acquisition stations in detail. In Section 3.2.2 the provided dataset is summarized. Finally, Section 3.2.3 discusses the advantages of the presented dataset.



### 3.2.1 Footwear Impression Acquisition

The goal was to create a dataset containing multiple footwear impressions for each shoe since this allows an evaluation of how well algorithms can compare footwear impressions under different conditions by also considering individual characteristics of a shoe sole influenced by wear and manufacturing. A trade-off had to be found to find the right balance for the time needed to acquire each impression. Thus, a mix of realistic but time-consuming and less realistic but less time-consuming impressions was chosen.



Figure 3.8: Acquisition line to collect footwear impressions.

An acquisition line with multiple consecutive acquisition stations was designed for an efficient acquisition workflow. As shown in Figure 3.8 the participants, who came with their own already worn shoes, were instructed to walk through multiple stations in order to leave different impressions of their footwear. As the goal was that the impressions should cover the complete tread patterns of the sole, the participants were instructed to “walk normally”, i.e., roll off their shoes and not just put them flat on the floor. Furthermore, since walking slowly on command is not that easy without losing one’s balance, walking sticks were provided. Figure 3.8 shows one of these acquisition stations. In the depicted example, the participants were asked to first apply oil on the soles of their shoes by stepping into a basket holding a cloth soaked in oil. After that, multiple impressions were produced by walking over sheets of cut wallpaper (in this case, with the left shoe). The impressions were labeled with the unique number assigned to each

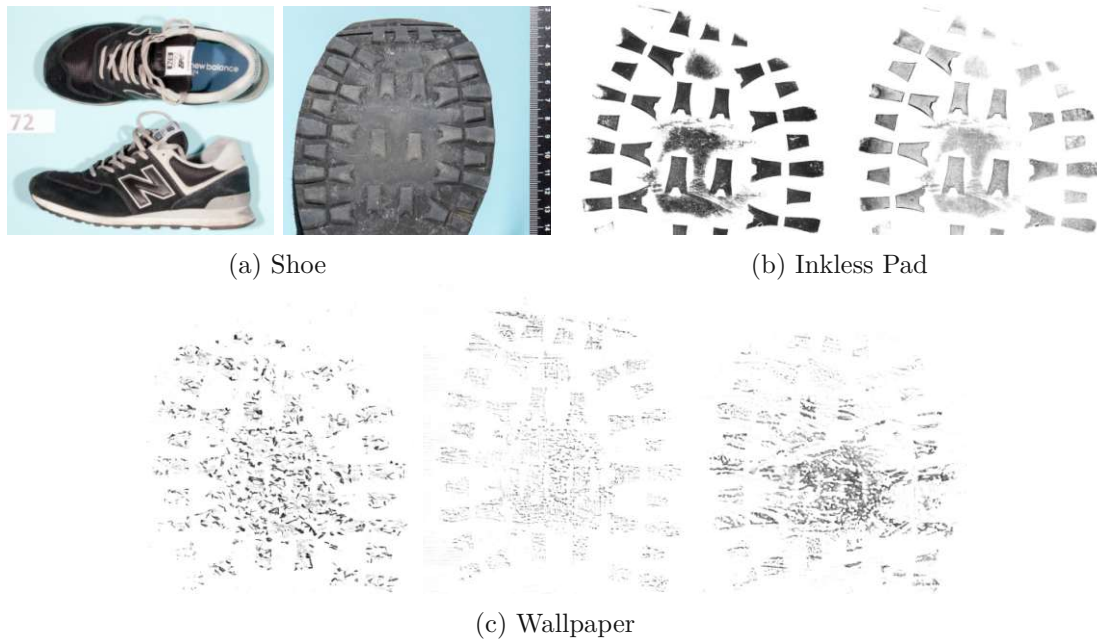


Figure 3.9: Images collected for each pair of shoes: composite shoe image and shoe sole (a), reference prints created using Inkless Pads (b) and oily impressions on wallpaper secured using black carbon powder (c).

pair of shoes at each acquisition station.

This section describes the acquisition stations and the decisions behind the chosen acquisition methods in detail.

### Metadata & Shoe

As a first step, to provide some general metadata, the shoe manufacturer and, if known, the shoe model were noted in addition to the shoe size. Even though this information is not enough to uniquely identify a specific tread pattern of a shoe, it still provides a rough reference. Subsequently, the shoes were photographed from the top and the side, as Figure 3.9 (a) shows in an example.

Since, in some applications, images of shoe soles are compared to impressions, the shoe soles were also photographed with a DSLR (Canon EOS 7D Mark II) mounted at a fixed distance. All images contain a reference frame to allow absolute measurements and comparisons to images from other acquisition methods. However, in contrast to the impressions made at the following stations, the shoe sole images still contain the natural curvature of the shoe, which has to be considered when comparing them to other images from the dataset.

### Inkless Pad

In many forensic applications, reference impressions are used for comparison since they provide high-quality images without noise in the background. In order to achieve that, two impressions were acquired for each shoe using Inkless Pads from the manufacturers Identicator Jacksonville and Neat Prints. One impression was made after stepping onto the Inkless Pad, and another one directly after that without re-applying the Inkless Pad solution to provide a more faded impression. The digitalization was done using a standard office scanner at 600dpi grayscale with all settings set to auto except sharpening turned off to prevent processing artifacts introduced by the printer software. In Figure 3.9 (b) resulting images are shown in an example. The second impression appears more faded than the first one due to the acquisition process described above.

### Wallpaper

As argued previously, the goal was to find a balance between realistic impressions and impressions that are fast and cheap to create to acquire a lot of different impressions quickly. For the latter, oily impressions on wallpaper were chosen since wallpaper can be found in a variety of different patterns in stores, can be purchased cheaply, and introduces noise to the impressions.

As shown in Figure 3.8, first, white mineral oil was applied to the shoe soles by stepping into a basket with a cloth soaked in such oil. After that, multiple impressions were made on sheets of cut wallpaper. Three different impressions for each shoe were created, i.e., six different impressions for each pair of shoes. For the first two impressions, wallpaper with regular patterns was chosen, which mostly stayed the same for the whole acquisition process, except for shortages in the end. For the third impression, alternating, more pronounced patterns were selected. The impressions were secured similarly to fingerprints by using carbon black powder (manufactured by BVDA), which sticks to the oily impression on the wallpaper. The impressions were digitalized by scanning with a standard office scanner at 600dpi, similar to the previous section. Figure 3.9 (c) shows crops of three different impressions from the same shoe created in this fashion. The first print shows a more pronounced thread pattern since the oil was applied only before the first and third impressions to produce a more faded second impression. It shows that the first two impressions contain a more regular noise pattern than the third impression due to the choice of wallpaper described above. In comparison to the reference impressions shown in Figure 3.9 (b), these images contain more noise, and individual characteristics like blemishes are not as clearly visible anymore.

### Realistic Impressions

Additionally to the wallpaper impressions described before, more realistic impressions were created. Since this process is time-consuming and expensive due to the cost of gelatin lifters, foam boxes, and others, only one of these more realistic impressions was

made for each pair of shoes. Depending on the last digit of the unique number given to each pair of shoes, different methods were used, which are listed in Table 3.5.

#	Method	Shoe
1	Newspaper	left
2	Cardboard	right
3	Styrofoam	left
4	Parquet	right
5	Perforated metal plate	left
6	Left + right overlapping	both
7	2 Pairs (7+8) overlapping	left
8	Smear	right
9	Wet	left
0	Foam box	right

Table 3.5: Summary of the realistic methods.

For the first eight methods (# 1-8), black gelatin lifters (manufactured by BVDA) and a roller were used to secure the dust patterns made by the shoes on the varying surfaces. This method was chosen since it is the most common method employed by the Austrian Police for securing footwear impressions at crime scenes. The gelatin lifters were then digitalized either using a photo-box with oblique illumination and a DSLR (Canon EOS-1Ds Mark II) or the Trasoscan by LIMS at 1000dpi. The impressions were made on various surfaces in order to capture a wide variety of background patterns. In detail, the following eight methods/surfaces were used:

**Newspapers** from 6 different Austrian publications since they provide a structured background to the impressions.

**Cardboard** with differently sized air chambers since they provide flexible impressions depending on the force applied. The air chambers of the cardboard also introduce a periodic pattern.

**Styrofoam** with thickness between 0.5cm und 1cm since it provides a regular pattern and also some flex.

**Parquet** since these floors are very common in Vienna. Chamfered edges were added to replicate the gaps between different tiles.

**Perforated metal plate** used as ceiling cover. Due to the wholes in the plates the impressions contain missing information and regular patterns.

**Left + right overlapping** impressions as overlapping impressions are very common at crime scenes. In this case impressions of both shoes of a pair are present. The impressions were made on a piece of paper.

**2 Pairs (7+8) overlapping** as overlapping impressions are very common at crime scenes. In this case impressions of this shoe pair and the next are present. The impressions were made on a piece of paper.

**Smear** impressions to capture impressions made with movement at the moment the sole touches the floor. The impressions were made on a piece of paper.

In addition to the methods described above, which were secured using gelatin lifters, **wet** (# 9) and **foam box** (# 0) impressions were made in the following way:

**Wet** impressions were chosen since they frequently occur at crime scenes and have a unique appearance. In order to create the impressions, first, a solution of 30g polyether sulfate dispersed in 170ml of distilled water was applied to the shoe soles using a sponge. The impressions were then made on a sheet of paper and secured using carbon black powder (manufactured by BVDA), similar to the wallpaper impressions described above. Digitalization was done again using the same scanner with the same settings as for the wallpaper impressions.

**Foam box** impressions provide a unique perspective since, in contrast to the other impressions, they are three-dimensional similar to the shoe soles themselves. These impressions were created by stepping into an orthopedic foam box (“Birkoschaum” manufactured by Birkenstock). Digitalization was done in two different ways. First, the foam boxes were photographed using a photo box with different oblique lighting directions to emphasize the 3D structure. Secondly, using plaster cast commonly used by dentists to create a 3D cast of the shoe sole. After drying and cleaning, the casts were also photographed similarly at varying lighting conditions.

Figure 3.10 shows crops of the digitized impressions above. For the structured surfaces like the newspaper, cardboard, styrofoam, parquet, and the perforated metal plate (# 1-3 & 5), the surface pattern is visible as background noise in the images desired. However, for parquet, the pattern is less regular. For the overlapping impressions (# 7 & 8), the challenge is to separate the impressions of the different shoes. The blurred impression (# 8) shows smeared individual characteristics which are only visible in parts of the image due to the shoe’s movement. Even though the wet impressions (# 9) are clearly separable from the background, they appear blurry. The 3D structure of the foam box (# 9) impressions provides an entirely different view of the tread patterns. Figure 3.11 shows that, as with other 3D impressions like toolmarks, different lighting conditions can change the resulting image significantly by emphasizing different parts of the impression. Even though it is omitted in the shown crops, all photographs include a scale in order to allow absolute measurements and comparisons.

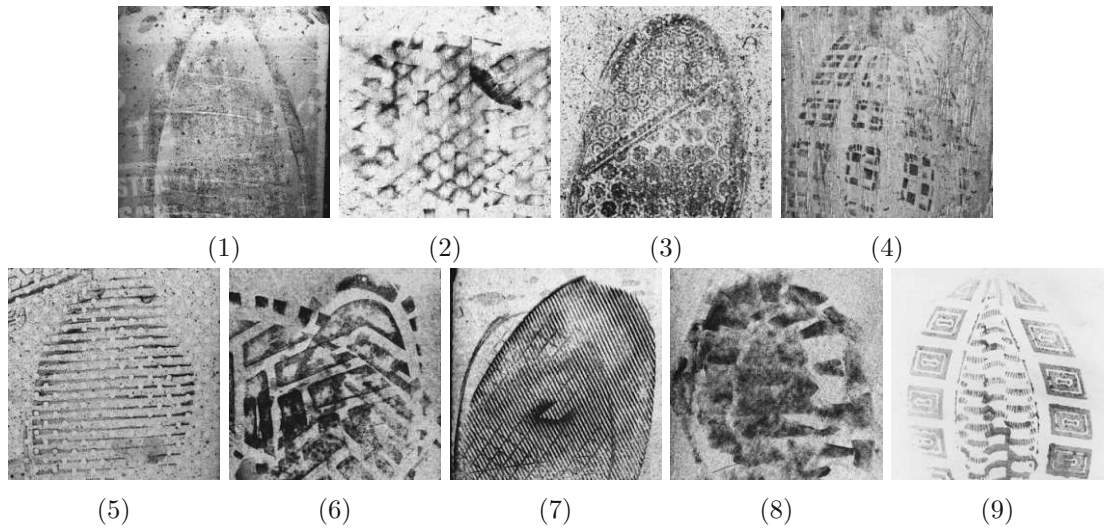


Figure 3.10: Realistic impressions collected depending on the last digit of the unique id for each shoe pair: Newspaper (1), Cardboard (2), Styrofoam (3), Parquet(4), Perforated metal plate (5), Left + right overlapping (6), 2 Pairs overlapping (7), Smeard (8) and Wet (9).

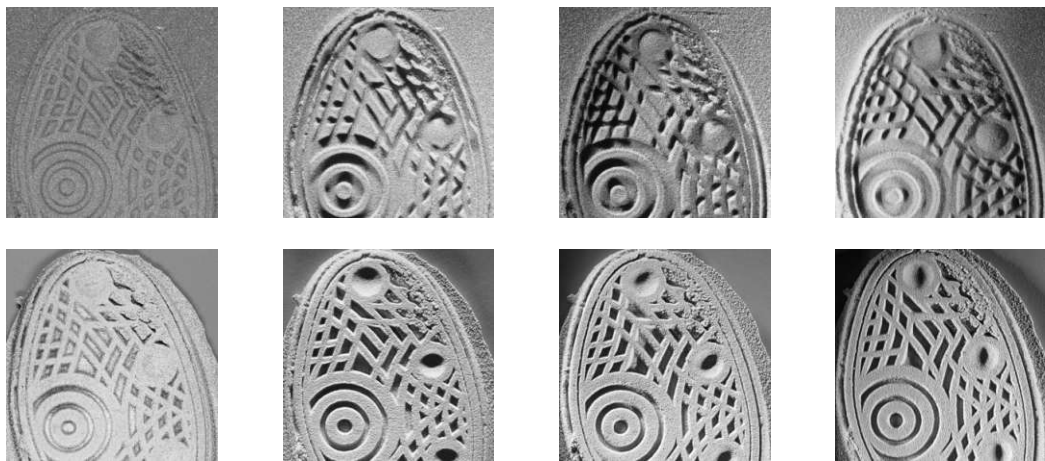


Figure 3.11: Foam box (top) and plaster cast of the same foam box (bottom) under varying illumination.

### 3.2.2 Dataset

The provided dataset<sup>4</sup>, which includes a total number of 300 different pairs of shoes, was created in two acquisition sessions in September 2018 and January 2019 with 69 and 231 different pairs of shoes, respectively. Since one goal is to allow the training of machine learning algorithms, it is suggested to use this partitioning for training (2019) and testing (2018) since this assures no accidental overlap. In summary, for each pair of shoes, the following images are provided:

- Composite shoe image top/side + shoe sole image
- Inkless Pad: 2 × left, 2 × right
- Wallpaper: 2 × regular wallpaper left + 2 × right; 1 × "special" wallpaper left + 1 × right
- Realistic: 1 × impression with at least one image depending on the method (last digit).

For every pair of shoes, the dataset provides 11 unique impressions with at least 13 different images with over 4,000 images in the whole dataset.

### 3.2.3 Conclusion

The presented dataset solves significant limitations of the current publicly available footwear impression datasets. First, it provides multiple impressions for each shoe to allow the training and evaluation of algorithms for comparing footwear impressions under varying conditions. Further, in contrast to the FID-300 dataset, a comparison of individual characteristics of footwear impressions is possible as it provides multiple impressions for each shoe. The high resolution of the images also supports such comparisons. In contrast to Richetelli et al.'s [RLL<sup>+</sup>17] high-resolution dataset, the dataset contains impressions of 300 different pairs of shoes. Additionally, impressions from both the left and the right shoe of each pair were acquired.

The diversity of the images provided allows the application in many different evaluation schemes. For instance, similarly to the FID-300 dataset, a crime scene vs. reference evaluation can be performed, comparing different reference images is possible, and even comparing shoe sole images with references and crime scene images can be evaluated. Furthermore, the wallpaper images could even be used to train a classifier, which can be tested using realistic impressions. This versatility sets this dataset apart from other publicly available footwear impression datasets.

<sup>4</sup><https://cvl.tuwien.ac.at/impress-dataset/>

## 3.3 Summary

This chapter introduced two new publicly available datasets with forensic images to overcome the limitations of available datasets in these fields.

Firstly, the toolmarks dataset FORMS was presented based on real break-ins from 48 different crime series investigated by the Austrian Police. Using a forensic microscope and a ring light with 11 different settings, 3,046 images were captured to investigate the influence of different lighting conditions on the accuracy of local similarity measures. An annotation tool was presented that allows efficient manual identification of matching points in the toolmark images to provide matching local image patches. Subsequently, the two ways the dataset is provided were described: annotated images and extracted matching and non-matching patches for a simplified quantitative comparison of local image descriptors.

Secondly, this chapter presented a novel footwear impression dataset, Impress, created to overcome significant limitations of other publicly available datasets in this field, like the lack of multiple impressions per shoe, resolution of the provided images, and the number of different shoes in the dataset. Similar to the FORMS dataset, a detailed description of the acquisition workflow is given. An acquisition line with multiple consecutive acquisition stations was designed to capture multiple footwear impressions of different modalities efficiently. Finally, the way the dataset is provided is described, and its advantages compared to other publicly available datasets are discussed.



# Methodology

In this chapter, the methodology proposed is presented in three domains introduced before that involve finding similarities between forensic images: toolmark analysis, writer identification and retrieval, and footwear impression comparison. Forensic experts traditionally make these comparisons by manually analyzing forensic material or images of such material. These experts are tasked to assess if two samples are significantly similar enough and thus have to be made with the same tool, written by the same author, or made with the same shoe. Even though this problem definition is similar for each of the domains in the focus of this thesis, the actual expertise needed to make this assessment is different. A handwriting expert is not trained to compare toolmarks, and a method developed to compare footwear impressions will not be able to determine if two handwritten pages are from the same author.

Thus, the methodology presented in this chapter proposes using a core trainable approach that is subsequently adapted to the specific challenges of each forensic domain individually. Generally, an automated system is desired by forensic experts that can quickly retrieve images from a database with thousands of forensic images similar to a query image. This way, the forensic experts can filter out many irrelevant images and concentrate their resources on ones that are most likely to be from the same writer, tool, or shoe. The goal is to help forensic experts identify matching forensic images more efficiently but not to make this decision for them. For such a system to be accepted by the end-users, the results must be accurate and returned as quickly as possible. Furthermore, such a system should not only work for forensic samples from known tools, shoes, or writers but should also be able to accurately retrieve similar samples from an unlabeled collection of samples that have never been used to train the system. Therefore, the proposed approach is based on metric learning with neural networks. As shown in Chapter 2 these methods allow the training of a robust embedding in which the similarity between new samples can be determined using a distance metric. Furthermore, instead of computing the similarity between each pair of images using a computationally expensive methodology, the samples

only have to be mapped once into the embedding, and the similarity is then computed efficiently using, for instance, the Euclidean distance.

These approaches have been shown to work with either whole images, like for comparing faces, or as local image descriptors that can be used similarly to traditional descriptors like SIFT. For forensic images, both the global context and fine-grained local image similarities are relevant to determining two samples' similarity. For example, a damaged area of a tool may leave a very distinct toolmark impression. When such a distinct shape is found in two toolmarks, it is highly likely, that the same tool made these toolmarks. However, there is still the possibility that two tools have this same distinct local characteristic just by chance. Thus, putting these local characteristics into a global context provides more robust evidence that the two toolmarks were actually made by the same tool or not. Similarly, an author's writing style is to a high degree defined by the way strokes are made on a level below the size of words or even characters. However, one stroke alone does not define an author's writing style but rather the statistical invariants present in these local characteristics, i.e., the reoccurrence of specific strokes.

The methodology proposed combines the local characteristics with a global context in two ways. Firstly, an end-to-end learning approach is proposed to let a CNN implicitly learn local image similarities and put them in a global context due to the inherent pyramidal structure. This approach is flexible since it only requires re-training to adapt to other domains. Nonetheless, it does not allow an explicit incorporation of domain-specific constraints and may require more training data to cover an unconstrained sample space. Secondly, another approach is presented that extracts local characteristics beforehand and trains a neural network using metric learning to describe their similarities efficiently. Subsequently, it puts them in a global context designed explicitly for the forensic domain it is applied to. Since, in this way, each image is divided into multiple local areas, this increases the number of samples that can be used for training, which makes this approach applicable in case creating a dataset with thousands of samples may not be feasible. Of course, an increased number of smaller training samples does not automatically mean that the information available to the neural network during training increases since the total number of pixels in the training set stays the same. Nevertheless, if the local characteristics are independent of the global context to a certain degree, the training samples can exhibit a wider variety. Consider, for example, a toolmark with the distinct local characteristics X at the beginning. The first layers of the neural network will learn this characteristic independently of the position on the toolmark since the receptive field is not wide enough to position it in the global context. However, the later layers, especially the final linear layer, will see X only at the beginning and may thus incorrectly learn that X can only occur at the beginning. Hence, uncoupling the local characteristics from the global context can lead to a representation that better captures the actual sample space. This strategy is particularly beneficial in the case of metric learning, where the neural network learns the similarity of multiple training samples. Coming back to the toolmark example, this may force the local characteristic X to be compared to another local characteristic Y during training which might otherwise never occur, in case X and

Y are never in the same position in the available training set.

However, this uncoupling has some disadvantages as well, and, as argued in Chapter 2, such handcrafted methods are often outperformed by end-to-end trained models in case the global context is a complex combination of local characteristics and choosing an appropriate model to represent it is not trivial. End-to-end based models circumvent this issue by providing enough samples, i.e., information, to learn an uninhibited representation with millions of parameters. Furthermore, as shown in Chapter 3, it is beneficial to provide the neural network with the information on what local image regions are similar, which is a time-consuming annotation task that has to be done by human experts, but may be, in many cases, more attainable than obtaining more forensic samples for training.

This section presents a machine-learning-based methodology, applicable, with adaptations, to diverse forensic domains. The proposed methodology is first presented for striated and impression toolmarks in Section 4.1. This section introduces the proposed metric learning approach based on triplet learning, which is used for all three forensic domains treated in this thesis. Subsequently, the proposed adaptations for writer retrieval and footwear impression comparison are discussed in Section 4.2 and Section 4.3, respectively. A detailed problem description is given for each forensic domain, and the modeling of both the local characteristics and the global context is described.

## 4.1 Toolmarks

Marks left by tools on surfaces are categorized as toolmark impressions, occurring when a part of the tool is pressed with force into the surface, and striation marks, which are created when a tool is moved over the surface, leaving a striated pattern. Figure 4.1 shows this on two examples. Inside the area where the tool had initial contact with the surface the tool's edge is impressed onto the surface. In these examples, the tool was moved over the surface with force. Therefore, they show striation marks in the direction of the movement. As a scale reference, in these examples, the width of the tool was below 6 mm.

As presented in Section 2.1.1, almost all publications on the automatic comparison of toolmarks apply to striated toolmarks. The toolmarks contain three different kinds of characteristics that can help identify the individual tool that created the toolmark [BKP<sup>+</sup>14]. Firstly, class characteristics are common to tools from the same class, which can identify the specific model of the tool. Secondly, sub-class characteristics then define more specific characteristics only shared by a subset of these tools. Examples of such characteristics are patterns in the material due to a specific machine used in the production of several consecutively produced tools. Thirdly, individual characteristics are specific to the individual tool. These characteristics are either similarly contributed by the production process or due to other influences like the wear. When a tool is used regularly, over time, parts of the surface chip away, and the surface changes due to abrasion. These characteristics distinctive for each tool are impressed onto the surface, and their visibility is highly dependent on the resolution of the imaging equipment. For instance, irregularities visible

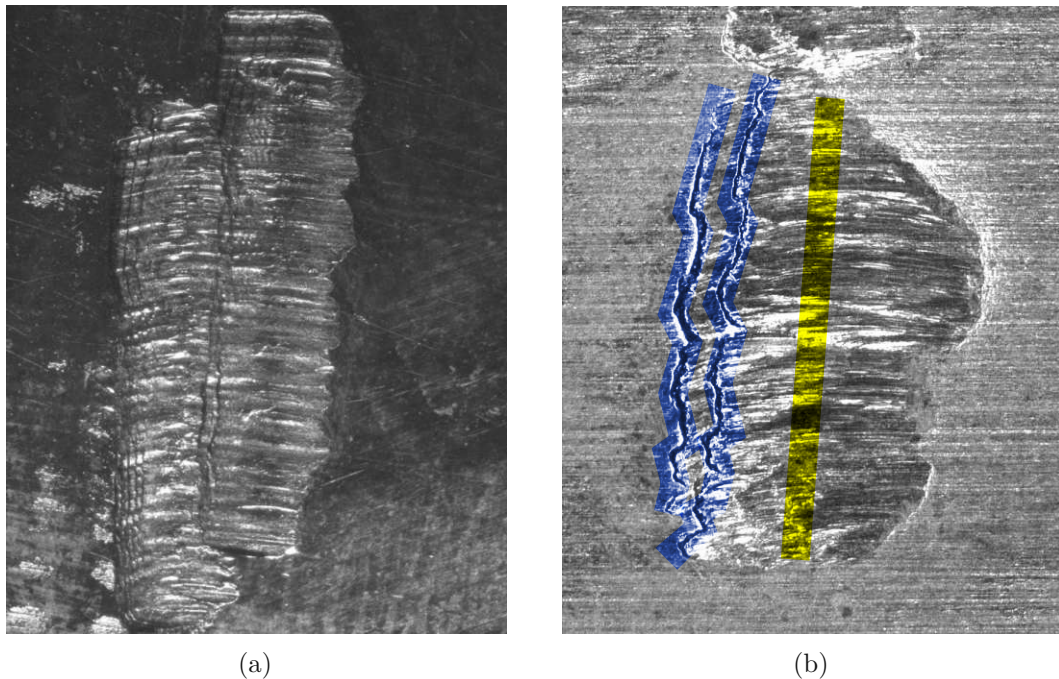


Figure 4.1: Example of toolmark images containing toolmark impressions and striated toolmarks. Since the edge of the locking pliers used in these examples was moved over the surface the initial contact of the tool leaves an impression, i.e. edge, and striated toolmarks in movement direction from left to right. In the right image the distinctly visible impressions are marked blue and the striation pattern in yellow.

at the impression marks in Figure 4.1 are most probably due to significant wear of the tool. Due to the limited resolution of these images, characteristics on a smaller scale are not visible in these examples. In particular striated toolmarks are often not perfectly visible in these low-resolution images. In general, toolmarks are captured using 2D imaging or 3D scans of the surfaces. Utilizing high-resolution 2D or 3D imaging, characteristics that may otherwise not be visible can be identified. In the example shown in Figure 4.2 the images were captured with a resolution of more than 400 pixels/mm, and the striated toolmarks were created under laboratory conditions. Thus, they show finer details than the examples in Figure 4.1, which show toolmarks left on locks during actual break-ins. 3D imaging techniques remove the uncertainty of varying lighting conditions and thus simplify the comparison of different toolmarks. However, there are variations due to the actual process of creating the toolmark, like the angle of attack, that can not be removed by using 3D imaging equipment. In Figure 4.2 five different striated toolmarks are shown made with the same tool (screwdriver) with varying angles of attack. There is significant variation when comparing toolmarks with a difference in angle of attack of 30 degrees or more, e.g., 15 degrees vs. 70 degrees which can be seen both in the 2D images and, more pronounced, in the superimposed 3D profiles.



Figure 4.2: Superimposed 1D profiles extracted from 3D surface scans onto 2D images of the NFI Toolmarks dataset [BKP<sup>+</sup>14]. All marks were made by the same tool with varying angle of attack; from top to bottom: 15 degree, 30 degree, 45 degree, 60 degree, and 70 degree.

Therefore, this section presents a methodology based on deep metric learning that learns these characteristics from examples. The proposed methodology is adapted to work with high-resolution 2D images of striated toolmarks captured under laboratory conditions and images of impression toolmarks of actual criminal cases with annotated toolmark edges. In contrast to other approaches focused on automated toolmark comparison like [BPZ15], as shown in Section 2, the methodology proposed uses 2D images and does not rely on 3D surface scans. Therefore, the approach can be applied to images captured under a forensic comparison microscope and does not need special 3D scanning equipment. Further, since the methodology is based on machine learning, examples with varying lighting conditions, surface materials, and other factors that influence the appearance of the toolmarks, like the angle of attack, can be integrated into the training set to tolerate such variations.

First, in Section 4.1.1, the extraction of the local characteristics is described. Since the training data available for both striated and impression toolmarks is limited, the way local characteristic are selected is crucial in these domains to learn an appropriate representation. Subsequently, Section 4.1.2 presents the triplet loss function and the network architectures used to attain this embedding representation. Finally, Section 4.1.3 shows how the embedding vectors, representing local characteristics, are put into a global context to yield a computationally efficient similarity metric for forensic toolmark images.

### 4.1.1 Extraction of Local Characteristics

Since the samples presented to the CNN during training define which features are learned, the selection of these samples is crucial. Firstly, the training set must contain enough samples from different classes to allow the CNN to identify the distinguishing characteristics. Secondly, class and sub-class variations must be represented to improve the robustness. Especially when dealing with small datasets, improvements can be made by either artificially increasing the variation or by carefully selecting the training samples (similarly to negative mining [BJTM16]). The presentation of the toolmarks' characteristics to the CNN during training defines which characteristics the CNN can use to learn to distinguish these toolmarks. By either augmenting the whole input sample or explicitly selecting parts of the input sample, it can implicitly be defined which characteristics of a sample are crucial for distinguishing multiple samples and which characteristics are ideally not be considered by the CNN. These artificial variations increase the robustness because the network learns that these variations are not distinctive. The proposed methodology is based on the triplet architecture, which tries to separate the positive samples from the negatives in the embedding. Since the weights of all three branches of the CNN are shared in this architecture, the feature extractors learned for all three samples in a triplet are the same; i.e., equal samples lead to equal representations in the embedding. These shared weights lead to the following reasoning: firstly, if a local characteristic is present in all three samples, this characteristic cannot improve the loss and is therefore suppressed by the CNN. Secondly, if a local characteristic is just present in the positive samples and not in the negative, the CNN uses this local characteristic to separate the samples. However, this does not only apply to the individual characteristics a CNN has to detect but also to unwanted characteristics like varying lighting conditions, small differences in camera angle, artificial data augmentations, and others. For instance, applying a data augmentation technique just to the positive samples would allow the network to find a trivial solution using exclusively artifacts caused by the augmentation. Since this minimizes the loss efficiently, this will cause a rapid decline of the loss function during training without learning a meaningful representation applicable to naturally occurring variations in the data.

As a result of these considerations, this section presents four different data augmentation strategies. These approaches are either based on augmenting the whole toolmark samples or extracting parts of the samples to uncouple the local characteristics from the global context to increase the number of training samples and provide a greater variety of different triplet combinations to the CNN. All the approaches described are applicable for striated toolmarks, which have a one-dimensional profile structure. For impression toolmarks, the local characteristics are two-dimensional; thus, only patch-based strategies are suitable. Even though toolmark impressions also have an underlying one-dimensional structure, since they represent the same toolmark edge as striated toolmarks, they require techniques to precisely identify this toolmark edge, which is not part of this thesis.

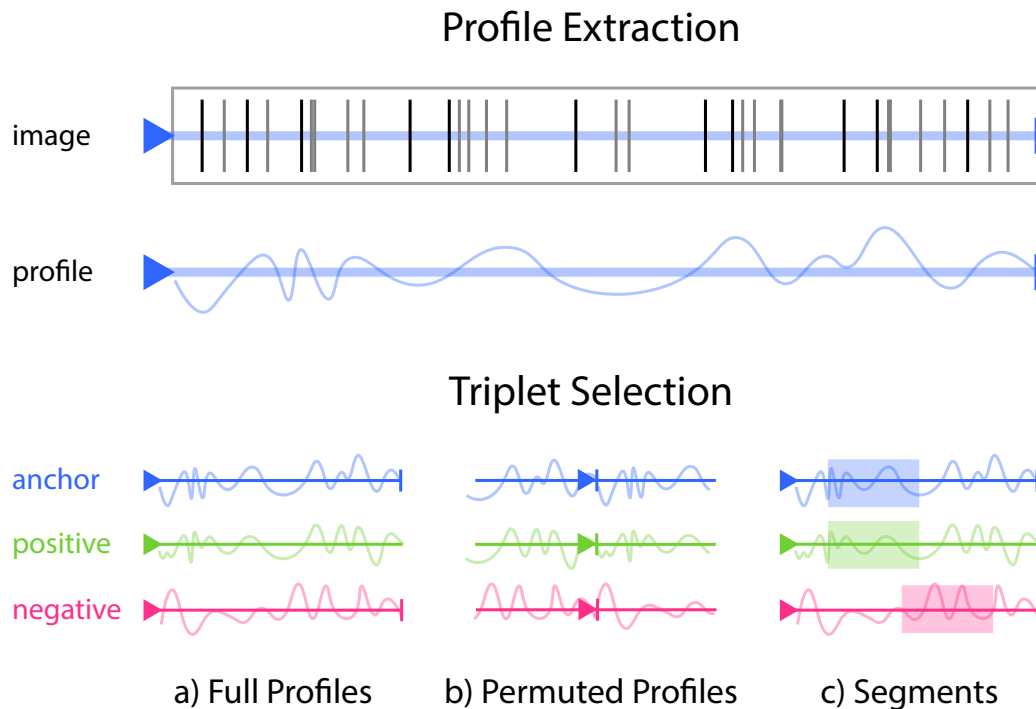


Figure 4.3: Profile extraction from striated toolmark images (top) and three different triplet selection strategies (bottom).

### Full Profiles

During training, at first, two toolmark images from the same tool and one toolmark from a different tool are selected. Subsequently, random vertical crops are taken from the striated toolmark images to increase the variability of the samples. Since, as shown in Figure 4.2, the movement direction is along an image axis, these vertical crops all describe the same toolmark profile. Nevertheless, since the extracted profiles are similar but not precisely the same, this should lead to a more robust representation of the striated toolmark. For evaluation, only center crops are used to ensure reproducibility. This approach is outlined in Figure 4.3a.

### Permuted Profiles

The training samples are chosen similarly to the full profiles by selecting two toolmark images from the same tool and one toolmark from a different tool and taking random vertical crops. However, all three samples, i.e., negative and positive, are additionally randomly permuted with the same factor as shown in Figure 4.2. Since the position of local characteristics is a defining factor for a tool, two new artificial tools are created during training with both matching and non-matching toolmark profiles in this way. The

permutation must be performed simultaneously on the whole triplet since if the negative sample is permuted independently, the position of the stitching artifact, i.e., the seam, can be learned by the CNN to distinguish positive and negative samples. As shown in Figure 4.2 the striated toolmark images exhibit a distinctly identifiable beginning and end, which amplifies this stitching artifact. Likewise, if all three samples are permuted independently, the exact position of a local characteristic on the profile is suppressed and can no longer be used to distinguish toolmarks. This strategy aims to increase the number of possible triplet samples by moving the individual characteristics to different positions and thus instructing the CNN to detect all possible local characteristics in the upper layers and observe these characteristics on all possible positions on the profiles. Like the full profiles, center crops without permutations are used for evaluation. Figure 4.3b sketches this approach.

### Segments

Similar to the full profiles, three toolmark images are chosen, an anchor, a positive, and a negative, and the profiles are extracted as random vertical crops from these images. However, instead of training with the complete profiles, profile segments are randomly cropped from the input images as a way to uncouple the local characteristics from the global context. This strategy extends the *Permuted Profiles* described above without introducing seams and can therefore be done for the positive samples and the negative sample independently, which increases the variety of combinations of local characteristics the CNN is presented during training. The approach is outlined in Figure 4.3c. As shown, the profiles of both the anchor and the positive sample are cropped at the same location, whereas another random segment is chosen from the negative profile sample. The overall architecture of the CNN branches is not changed for this. However, since the number of pixels in the input samples is reduced, fewer parameters are needed for the fully-connected layers. The length of the segments is evaluated in Section 5.2. Of course, uncoupling the local characteristics from the global context means that it has to be modeled separately, which is described later on in Section 4.1.3.

### Patches

The preprocessing pipeline proposed by Baiker et al. [BKP<sup>+</sup>14] for the striated NFI Toolmark profiles includes an averaging along the x-axis. Therefore, random square patches are cropped from the input images to investigate the influence of adding a second dimension for noise reduction. Similar to the profile segments, the positive samples are cropped simultaneously, i.e., at the same position. However, preliminary tests showed that the negative sample must be extracted from one of the positive images to prohibit trivial solutions like different lighting conditions, contrast, and others. A safety distance ensures that the negative sample does not overlap with the positives. Since the NFI Toolmark images only contain the cropped toolmark area, negative samples extracted this way are guaranteed to show part of the toolmark and not just the background. Therefore, in this case, no special considerations are needed to avoid training a background/foreground



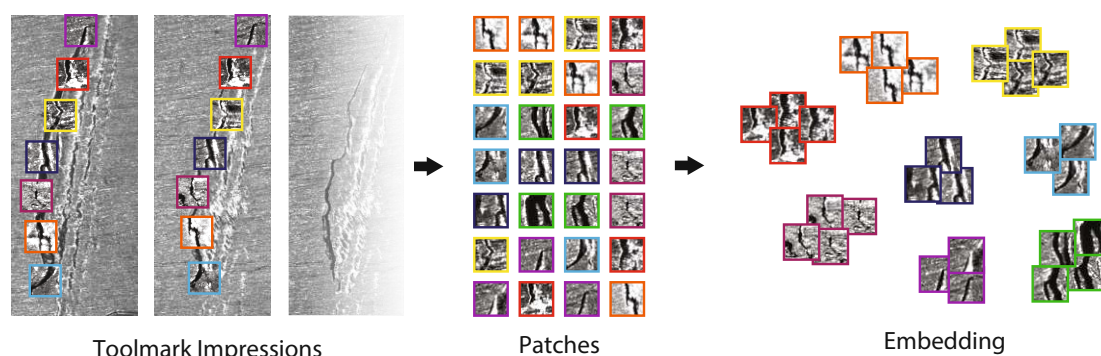


Figure 4.4: Extracting patches from the toolmarks and mapping them into an embedding using a triplet network.

classifier by accident. Furthermore, the samples are randomly flipped horizontally, i.e., reflected along the central vertical axis, to counteract the learning of slight angular variations in the camera angle. As with the profile segments, sliding windows in the center of the toolmark images are used to compute similarity scores during evaluation.

For impression toolmarks, the toolmark edge cannot simply be represented without precisely identifying it in the images, which is not part of this thesis. Therefore, the patch-based approach for striated toolmarks is adapted for impression toolmarks. Contrary to the approach described above, the patches are utilized to directly represent the local characteristics instead of just reducing the influence of noise. Since, as shown in Figure 4.4, these local characteristics are not always located on a straight line, the patches are extracted along the annotated polylines provided by the FORMS dataset presented in Section 3.1. That way, the patches can be treated as described above. This method can be seen as a two-dimensional extension of the segment-based approach above. Nevertheless, as the patches are used to encode complex two-dimensional local characteristics, different augmentation and sample selection strategies are evaluated in Section 5.3. That includes introducing random rotations to reduce the influence of angular differences in the annotation process and just matching patches from different lighting settings in the same image location to remove the influence of the annotation process altogether. In contrast to the NFI Toolmark images, the toolmark edge is only part of the whole lock cylinder in impression toolmark images from the FORMS dataset. Therefore, patches for the negative samples are also extracted from the annotated polylines to ensure they contain toolmark characteristics to avoid trivial solutions. Furthermore, the negative patches are extracted from the positive images, similarly to the NFI Toolmark images. A safety distance guarantees that the randomly chosen negative patch lies on a point on the polyline that is different from the positive samples, i.e., that the negative patch does not show the same local characteristics.

### 4.1.2 Triplet Learning

The proposed metric learning methodology is a triplet based network. The training is performed by forwarding three input samples (a triplet  $T = \{x_{p_1}, x_{p_2}, x_n\}$ ) through three identical neural network branches, i.e. they are mapped into the embedding  $f(x_i)$ .

Similar to siamese networks [CHL05], the network architecture consists of multiple branches with shared weights, as shown in Figure 4.5. The training is performed, by forwarding three input samples, i.e. a triplet, through these equal CNN branches. Each triplet consists of an anchor  $x_{p_1}$ , a positive (matching) sample  $x_{p_2}$  and a negative (non-matching) sample  $x_n$ . The results are then combined in the loss function, and the error is back-propagated subsequently.

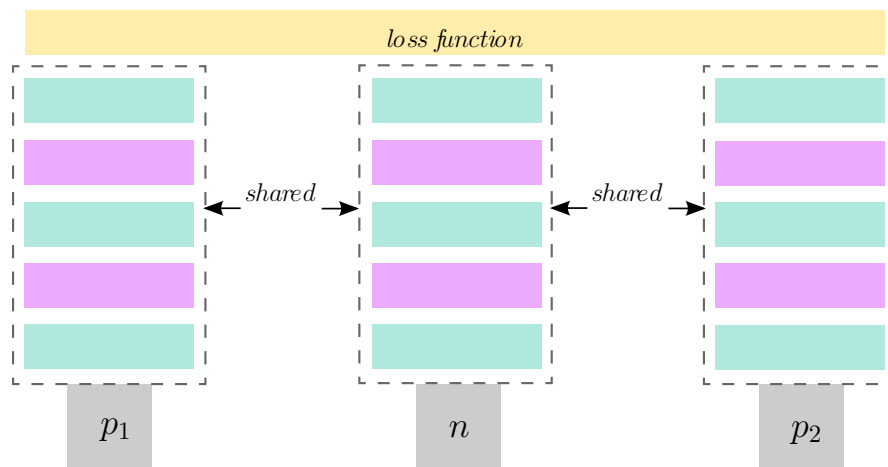


Figure 4.5: Triplet architecture

The dimension of this embedding  $f(x)$  can be controlled by changing the size of the last layer in the branches. Since the weights are shared only one branch is needed after the training. The loss function minimizes the Euclidean distance between matching samples and therefore the  $L_2$  norm can be used to measure distances in the embedding. Consequently, efficient algorithms for calculating  $L_2$  distances can be applied [BJTM16]. Additionally, the storage requirements are directly controlled by changing the dimension of the embedding.

The use of two different loss functions is proposed. On the one hand, the probabilistic SoftPN loss based on the work of Balntas et al. [BJTM16], which uses the ratio between the exponential distances and is defined as:

$$\ell_{ratio}(T) = \left( \frac{e^{\Delta^+}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 + \left( 1 - \frac{e^{\Delta^*}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 \quad (4.1)$$

with the distances between the three samples in a triplet:

$$\begin{aligned}\Delta^+ &= \|f(x_{p_1}) - f(x_{p_2})\|_2 \\ \Delta_1^- &= \|f(x_{p_1}) - f(x_n)\|_2 \\ \Delta_2^- &= \|f(x_{p_2}) - f(x_n)\|_2\end{aligned}\tag{4.2}$$

and  $\Delta^* = \min(\Delta_1^-, \Delta_2^-)$ . The second loss utilized is an extension of the margin loss used for siamese networks for triplets and is also proposed by Balntas et al. [BRPM16]. The loss tries to force  $\Delta^+$  to be smaller than  $\Delta^*$  by a margin  $m$ , set to 1.0 by default. The loss is defined as:

$$\ell_{margin^*}(T) = \max(0, m + \Delta^+ - \Delta^*)\tag{4.3}$$

In other triplet formulations, e.g., used by the PyTorch framework<sup>1</sup>, the triplet is defined by an anchor  $x_a$ , a positive sample  $x_p$  and a negative sample  $x_n$ . In this definition, computing  $\Delta^*$  corresponds to an anchor swap, which swaps the anchor with the positive sample in case its distance is closer to the negative sample, and the margin loss is defined as:

$$\ell_{margin}(T) = \max(0, \Delta^+ - \Delta_1^- + m)\tag{4.4}$$

with  $x_a = x_{p_1}$ ,  $x_p = x_{p_1}$  and the distances as defined above. Without performing the anchor swap, the negative distance  $\Delta_1^-$  is not used for computing this loss function, and therefore, this loss function might lead to inferior performance, as argued by Balntas et al. [BRPM16].

The proposed loss functions can be combined with different network architectures, which allows an adaption to different domains. Even though an exhaustive comparison of network architectures is not the focus of this thesis, shallow traditional CNNs are compared to ResNet [HZRS16] and ResNet-like architectures, which include modern improvements like skip connections and batch normalization. The use of shallow architectures has two reasons: firstly, Balntas et al. [BJTM16] showed that such architectures adapt to different domains and outperform traditional feature descriptors for comparing local image patches. As the presented methodology focuses on uncoupling the global context from the local characteristics, such shallow networks, like the PN-Net by Balntas et al. [BJTM16] designed for comparing local image patches, are well suited. Secondly, simpler models are less prone to overfitting. Even with the data augmentation techniques described in the previous section, the available number of training samples is limited for the forensic domains discussed in this thesis. The NFI dataset, for example, contains only 300 striated toolmarks from 50 different tools. As such, using less expressive models can benefit the problems discussed in this thesis. Nevertheless, publications like Christlein et al. [CGFM17] show that if enough training data is available (e.g., 480k  $32 \times 32$  patches), ResNets are suitable for encoding local characteristics. Thus, modern deep architectures are employed for writer identification, toolmark impressions, and footwear impressions, where the number of training samples is less restricted. However, to avoid overfitting,

<sup>1</sup><https://pytorch.org/docs/stable/generated/torch.nn.TripletMarginLoss.html>

for encoding local image patches, the DenseNet architecture is used since it utilizes the parameters more efficiently than ResNets [HLvW17].

In the following sections, three different network architectures for the triplet branches are presented, starting with the original PN-Net proposed by Balntas et al. [BJTM16], which they show works well for comparing local image patches. Subsequently, a similarly shallow architecture, TripNet, based on the PN-Net and adapted for learning the similarity of striated toolmarks, is shown. Finally, a deep network based on the DenseNet proposed by Huang et al. [HLvW17] is presented, which improves on the previous networks and is well suited for the challenging comparison of local image patches of toolmark impressions.

### PN-Net

The architecture proposed by Balntas et al. [BJTM16] is designed to compute descriptors for  $32 \times 32$  patches efficiently and thus only contains two convolutional layers. One pooling layer is used for downscaling and Tanh as non-linear activation functions. A final fully-connected layer collapses all dimensions to a 128- or 256-dimension embedding vector. Table 4.1 shows the proposed network architecture in detail. Interestingly, even though the architecture is very shallow with only two convolutions, the convolutional kernels with a size of  $7 \times 7$  and  $6 \times 6$  allow for a wide receptive field.

Layer #	Description
1	SpatialConvolution(7,7) $\rightarrow$ 32
2	Tanh
3	MaxPooling(2,2)
4	SpatialConvolution(6,6) $\rightarrow$ 64
5	Tanh
6	Linear $\rightarrow$ nfeat $\in$ {128, 256}
7	Tanh

Table 4.1: PN-Net architecture of the PN-Net CNN branches [BJTM16]

### TripNet

The architecture of the CNN is depicted in Table 4.2. In case profiles (or profile segments) are used as input samples, the convolutional and pooling layers have one-dimensional input regions. For patches, the architecture of the CNN branches is changed accordingly, i.e.,  $5 \times 5$  and  $3 \times 3$  regions are used for the convolutional and max-pooling layers, respectively.

Batch normalization [IS15] follows each convolutional layer to decrease the dependency on input normalization and initialization of the network. The size of the convolutions, the number of feature maps, and the size of the pooling layers were empirically evaluated. The best results are achieved with  $1 \times 5$  convolutions and  $1 \times 3$  pooling with 64 feature maps in the first convolution and 32 in the second. In contrast to the original PN-Net described

above, Rectified Linear Units (ReLU) [LBOM98] and average pooling [LBBH98] are used since this setup performs more desirable. However, a Tanh activation function ensures a smooth output of the last layer. To additionally fight overfitting due to the relatively small dataset used, a Dropout [HSK<sup>+</sup>12] layer is added at the end with a probability of 0.5.

Layer #	Description
1	SpatialConvolution(1,5) → 64
2	SpatialBatchNormalization
3	ReLU
4	AveragePooling(1,3)
5	SpatialConvolution(1,5) → 32
6	SpatialBatchNormalization
7	ReLU
8	AveragePooling(1,3)
9	Dropout
10	Linear → nfeat
11	Tanh

Table 4.2: Architecture of the TripNet CNN branches.

## Deep Network

In contrast to the previous shallow networks based on the PN-Net proposed by Balntas et al. [BJTM16], this section presents a deep architecture based on the DenseNet [HLvW17]. Even though it is similar to ResNets [HZRS16] with its skip connections, DenseNets features maps are concatenated instead of summed. Huang et al. [HLvW17] argue that by connecting each layer to every other layer in a dense fashion to facilitate maximum information flow, deeper network architectures can be trained efficiently to attain more accurate models. Therefore, the available parameters are used more efficiently since features can be reused throughout the network [HLvW17]. In their evaluation, Huang et al. [HLvW17] achieve comparable results to ResNets with 1/3 of the parameters. Furthermore, they show that DenseNets are less prone to overfitting, particularly useful for forensic domains, like toolmarks, with little training data.

The dense blocks combine features by concatenating the feature maps of all preceding convolutional, which introduces  $\frac{L(L+1)}{2}$  connections for a network consisting of  $L$  layers. The growth rate  $k$  defines how many are added by each layer. Each layer in a dense block consists of batch-normalization, followed by ReLU activations and  $3 \times 3$  convolutions. Figure 4.6 shows such a dense block with five layers and a growth rate of 4 and the connections between the layers. Introducing  $1 \times 1$  convolutions as bottleneck layers before the  $3 \times 3$  convolutions reduces the number of feature maps for computational efficiency. This architecture achieves downsampling by adding transition layers after each dense

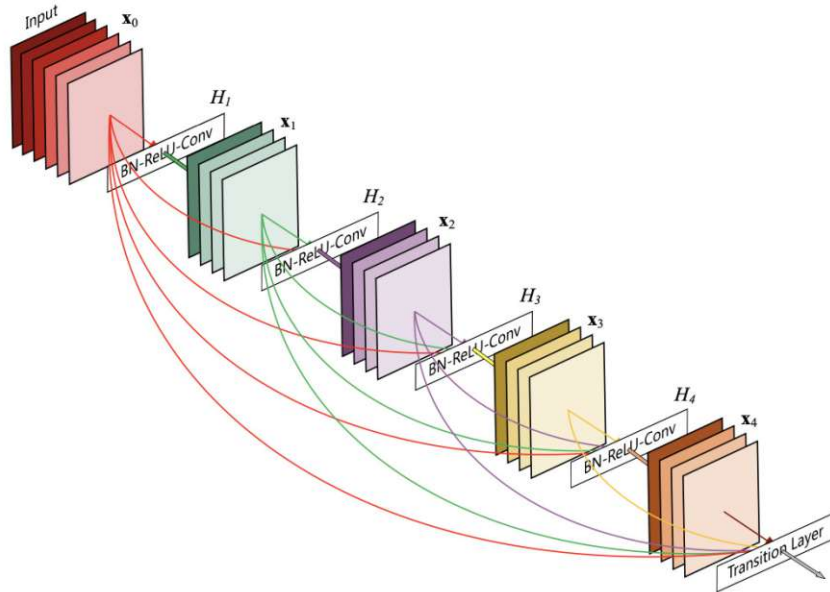


Figure 4.6: 5-layer DenseNet block visualizing the dense connections between the convolutional layers.

block with  $1 \times 1$  convolutions and  $2 \times 2$  average pooling. Similar to the bottleneck layers, a compression layer with a reduction factor set to 0.5 per default can be employed to reduce the number of feature maps generated by the transition layers. Since DenseNets are proposed to replace the shallow CNNs described above, both the compression and bottleneck layers are utilized to maximize the model's compactness. Analogous to the PN-Net and the TripNet, a Tanh activation function is used for the output of the last layer to ensure a smooth surface for the learned embedding.

### 4.1.3 Encoding Global Context

Since the  $L_2$  distance is utilized in the loss function described above, similarity scores for embedding vectors can be computed using the Euclidean distance. For striated toolmarks, each toolmark is represented by one embedding vector in case full profiles and permuted profiles are utilized. For these representations, the global context is encapsulated by the CNN and the similarity (or dissimilarity) of two toolmark samples  $x_1$  and  $x_2$  is given by the  $L_2$  distance of their corresponding embedding vectors  $f(x_1)$  and  $f(x_2)$ , respectively:

$$d(x_1, x_2) = \|f(x_{x_1}) - f(x_{x_2})\|_2 \quad (4.5)$$

In case of profile segments or patches, the embedding vectors represent local characteristics that have to be combined in a separate step to calculate the similarity between two toolmarks.

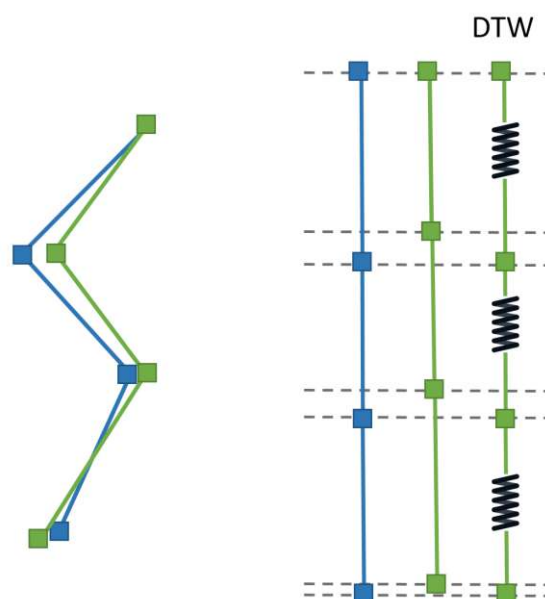


Figure 4.7: Matching in fixed step sizes compared to a distance computation using dynamic time warping (DTW).

For this two different approaches are proposed: a sliding window approach and dynamic time warping (DTW).

The sliding window approach uses the embedding representations of the profile segments or patches and computes the  $L_2$  distance for each pair of segments or patches from top to bottom. The sum of the pairwise distances is then used as the distance measure between two toolmarks. Since the striated toolmarks from the NFI dataset have the same length, no alignment is necessary, and only the step size has to be defined as a parameter beforehand, which is set to  $1/16$  of the height of the segment or patch by default to provide enough overlap. The impression toolmarks provided by the FORMS dataset have varying lengths and contain both complete and partial toolmarks. Therefore, an alignment strategy is employed for these samples. For all possible alignments, the similarity is computed as described above for the striated toolmarks. The minimal distance is then taken as the similarity measure between the two toolmarks, which is additionally normalized by the length to provide comparable results.

The advantage of this sliding window approach is that it is simple and computationally inexpensive. However, it requires accurate annotations or images created under constrained laboratory conditions, such as the striated NFI toolmarks, since local characteristics on different parts of the toolmark may be compared against each other otherwise. In particular slight variations in the angles can lead to accumulated length differences that cannot be compensated, as shown in Figure 4.7, which is a problem in particular for the manually annotated FORMS impression toolmarks. Therefore, to relax the fixed step size between two local characteristics, a DTW (Dynamic Time Warping) [BC94]

approach is proposed to allow more flexible matching. Using DTW, minor inaccuracies in the annotation process, and the resulting changes in the length of the toolmark segments, can be compensated. Figures 4.7 visualizes the advantages of the DTW approach.

## 4.2 Writer Retrieval

For writer retrieval and identification, the writing style of a handwritten page has to be efficiently encoded to facilitate fast retrieval of similar handwritings, i.e., handwritings by the same writer. Since the goal is to identify the writer independent of the content of the handwritten text, it is crucial that this representation encodes just the writing style and not the content of the written text. Therefore, other approaches for writer retrieval, like Fiel and Sablatnig [FS13], utilize a similar uncoupling of local characteristics from the global context, as proposed above for toolmarks, by using local descriptors like SIFT to describe handwritten strokes and combining these features using an encoding like the Fischer Vector. The idea is to remove the positional relations of the local characteristics representing words on the handwritten page and utilize their distribution to encode the writing style. Similar to other methods shown in Chapter 2, as for instance [CM18], the methodology proposed utilizes CNNs to encode the local characteristics of the handwriting. However, instead of using a network for classification, a methodology based on metric learning is proposed, which utilizes a triplet network that learns a similarity measure for image patches. Patches are extracted from the handwriting and mapped into an embedding where the  $L_2$  distance defines their similarity. The similarity measure is thus learned directly from the handwriting, which represents the writing style. This mapping can then be used like traditional features for image patches by encoding each image patch using the learned feature descriptor. The global context of the handwriting is captured by encoding these features for each document image using either the Vector of Locally Aggregated Descriptors (VLAD), like in Christlein et al. [CGFM17], or a Fisher Vector, like in Fiel and Sablatnig [FS13].

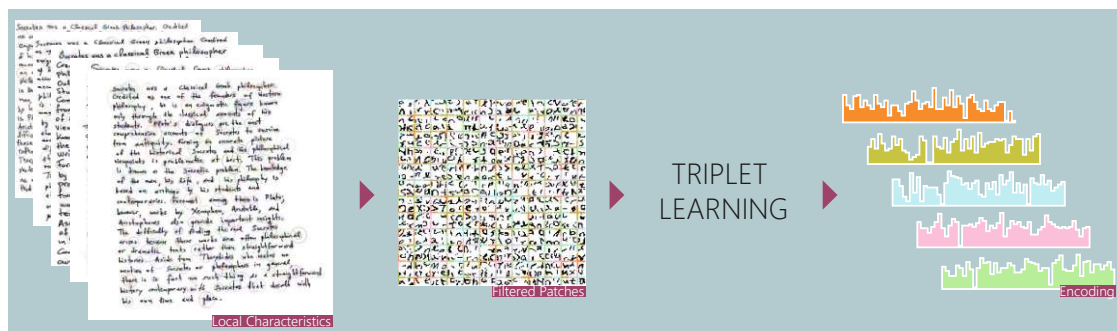


Figure 4.8: Overview of the writer retrieval methodology. The local characteristics are extracted from the handwritten page, filtered, and subsequently used to train a CNN with a triplet loss. The distribution of the local characteristics of each handwritten page is then encoded using VLAD or the Fischer Vector.



An overview of the methodology proposed is given in Figure 4.8. In the following sections, the methodology proposed is described in detail. First, the images are binarized and local characteristics, i.e., image patches, are extracted. These patches are then presented to the network, which learns a mapping based on these patches, minimizing intraclass distances and maximizing interclass distances. The mapped representations obtained by the network are then used to generate an encoding of the writing style. Whitening is applied as a post-processing step to limit the impact of visual word co-occurrence. The comparison of pages can then efficiently be performed by comparing the encodings of the respective pages.

This section is divided as follows: first, the extraction of the local characteristics is explained, and subsequently, the similarity measure learned using triplet learning. After that, for the encoding of global context, the Fischer Vector and VLAD are presented, and whitening of the data is described.

#### 4.2.1 Extraction of Local Characteristics

The method takes a binarized image of a handwritten page as input. Binarization is not necessary for the rest of the pipeline, but since some databases only provide binarized images, this step was introduced using Otsu’s method [Ots79]. Furthermore, the background should not influence the learning of the features, and binarization is a simple way of removing the background. However, separating the handwriting from the background is challenging for some documents like historical data and is not in the scope of this work. Therefore, the methodology proposed assumes that the handwriting is clearly separable.

In this work, two different approaches for extracting the local handwriting characteristics are proposed. Firstly, the extraction of image patches randomly with a defined percentage of handwriting present within. This threshold is set to 15% black pixels within each extracted patch to filter out patches containing only dots, punctuations, or individual partial strokes but otherwise does not restrict the extraction. Secondly, the locations of SIFT keypoints, which originate from the Harris Corner detector, are used as the center of the patches. The advantage of random patches is that a fixed number of patches can be set for extraction, and thus each image contains the same number of patches. Additionally, this way, the extracted patches exhibit a more significant variance, allowing the triplet network to see a more diverse selection of patches. The drawbacks are that it can take longer to find the patches that contain enough information since the random location may lie between lines or at the ending of a stroke. Furthermore, if only a few words are present in the document image, the patches are extracted from similar locations. Thus, the information is redundant, which may lead to performance loss because unusual characteristics of a specific writer are over-represented. When using SIFT keypoint locations, the advantage is that previous methods, such as [FS12] and [FS13], have shown that there is enough information around these locations for a successful identification or retrieval, and further, these keypoints lie on or near the strokes. They also show that even though the number of keypoints varies heavily, this

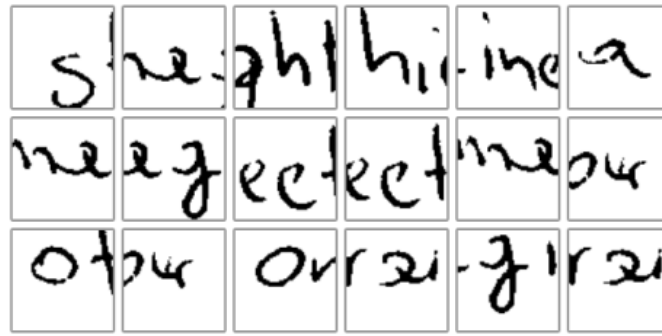


Figure 4.9: Sample patches extracted at the SIFT keypoint locations.

has no negative influence on the performance. The size of the patches defines how much context the triplet network utilizes. However, the general idea is that the patches should represent the local characteristics of the writing style and not the written words or even whole sentences. Thus, two different sizes are evaluated. One strategy extracts patches showing whole characters and their transitions and another showing sub-character level strokes;  $64 \times 64$  pixels and  $32 \times 32$  respectively. In Figure 4.9 sample images, patches that have been extracted at the SIFT keypoint locations are shown. In this case, one to three characters are shown depending on the handwriting size and the location.

### Surrogate Classes

In [FS12] and [FS13] the SIFT features are filtered according to their size. The idea is to ignore the features with small and large sizes since they are mostly located at the end of a line, of a character, or between text lines. [CGFM17] use the SIFT features to filter the patches after the creation of the surrogate classes, i.e., the clustering. They use the distance ratio of the two distances between the closest and second-closest cluster center. This approach filters out patches that lie between clusters and are thus not representative of any particular class.

The methodology proposed adopts such an approach to filter out patches in the training step. However, a lower number of classes is used to get character-like clusters, e.g., 100 clusters. The goal is to filter out patches with patterns that do not occur often and therefore do not form a cluster. This filtering is restricted to the training step since this might filter out patches containing writer-specific features during evaluation. The reasoning is that the system should learn to distinguish between different writers within these clusters. Clustering is performed using k-means, and patches with a distance ratio of 0.9 or more between the nearest cluster and the second-nearest cluster are removed, as proposed by Christlein et al. [CGFM17].

### 4.2.2 Triplet Learning

Even though other approaches, like [FS15], have shown that using the output of the last layer of a fully connected CNN trained for classification, e.g., with a Softmax layer and a Mean Square Error loss function, work well in this work directly training an embedding from triplets of patches is proposed. Similarly to toolmarks the triplet loss by [BJTM16] described in Section 4.1.2 is used.

Two different CNN architectures are utilized for learning handwriting similarities in this work. Firstly, a shallow architecture with just three convolutional layers and one linear layer, as depicted in Table 4.3. This architecture is similar to the original architectures proposed by Balntas and the TripNet proposed in Section 4.1.2. Batch normalization is performed after each convolutional layer to decrease the dependency on input normalization and initialization of the weights [IS15]. The total number of layers, feature planes of the convolutional layers, and filter sizes for the convolutional and MaxPooling layers were determined experimentally. Similarly to [BJTM16] Rectified Linear Units (ReLU) are used. Since it has shown no negative effects, Dropout with a probability of 0.5 is used in the last layer to avoid potential overfitting. A smooth output is ensured by applying Tanh activation functions in the last layer.

Layer #	Description
1	SpatialConvolution(7,7) $\rightarrow$ 32
2	SpatialBatchNormalization
3	ReLU
4	MaxPooling(3,3)
5	SpatialConvolution(5,5) $\rightarrow$ 64
6	SpatialBatchNormalization
7	ReLU
8	MaxPooling(3,3)
9	SpatialConvolution(5,5) $\rightarrow$ 64
10	SpatialBatchNormalization
11	ReLU
12	MaxPooling(3,3)
13	Dropout
14	Linear $\rightarrow$ nfeat
15	Tanh

Table 4.3: Architecture of the CNN branches

Secondly, a DenseNet architecture similar to the one proposed for the toolmarks in Section 4.1.2 is utilized. This architecture uses a total number of 50 layers with a growth rate of  $k = 12$  and three blocks.  $1 \times 1$  convolutions are used as bottleneck layers to compress the number of channels, as proposed by Huang et al. [HLvW17].

For both architectures, the output of the last layer determines the embedding dimension. As in [CGFM17] it is set to 128. However, additionally embedding dimensions of 32, 64 and 256 are evaluated to investigate the influence of this parameter.

### 4.2.3 Encoding Global Context

The patches extracted from the document image are mapped into the embedding learned by the CNN. Their representations are then encoded to form a feature vector for each document image. For this encoding of the global context, two methodologies based on visual words are proposed, which encoded the occurrences of visual features. Firstly, the Fisher Vector, which was proposed with SIFT features by Perronin et al. [PD07][PSM10] as an improvement for the Bag Of Words (BOW) method, which has also been successfully applied to writer retrieval and identification by Fiel and Sablatnig [FS13]. Secondly, the VLAD encoding [JPD<sup>+</sup>12] a simplified non-probabilistic version of the Fisher Vector, which has also been successfully applied to writer retrieval and identification by Christlein et al. [CBA15]. It outperforms the BOW methods and provides comparable results to the Fisher Vector [JPD<sup>+</sup>12]. In this section, both methods are described in detail.

#### Generation of the Fisher Vector

The BOW method uses k-means for clustering the feature space, and when identifying a new document image, an occurrence histogram of the nearest cluster center is generated. In contrast to this, when following the method of Perronin et al. [PSM10], a Gaussian Mixture Model (GMM) is used for clustering the feature space, and higher-order statistics are exploited to generate a feature vector. This approach has the advantage that the separation of the feature space is not as strict as when using k-means, which especially affects features that lie nearly in the middle of two or more cluster centers. When only counting the occurrences of the nearest cluster center, the fact that a specific feature point is also close to another center is ignored. The feature space can be described more precisely using a GMM and higher-order statistics. Figure 4.10 shows the separation of a simplified feature space. The dashed lines represent the separation when applying k-means, and the colors illustrate the separation when using a GMM. The location of the features within a cluster does not have any influence when counting the occurrences of the nearest cluster center. However, if GMM statistics are used, the locations of the features do matter.

After the GMMs are fitted to the training data, the feature vector of a document image can be generated. This feature vector, comprising the mapped image patches  $\mathcal{X} = \{f(x_t), t = 1 \dots T\}$  where  $f$  is the mapping function learned by the CNN, is computed by [FS13]:

$$\mathcal{G}_k = \frac{1}{\sqrt{w_k}} \sum_{t=1}^T P(k|f(x_t)) \left( \frac{x_t - \mu_k}{\sigma_k} \right) \quad (4.6)$$

where  $\mathcal{G}_k$  is the feature vector for one specific distribution  $k$ . The weights of the  $k$ -th distribution are given by  $w_k$ , and  $\mu_k$  and  $\sigma_k$  are the means and the variance of the

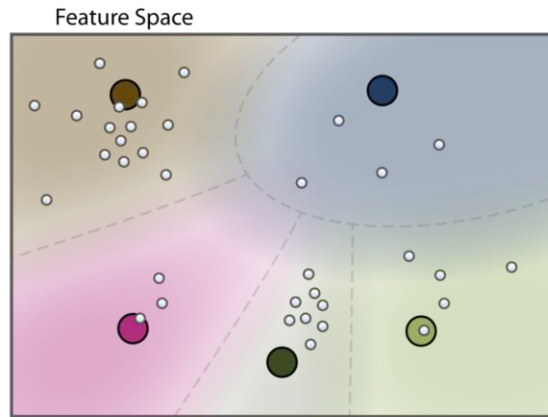


Figure 4.10: Separation of the feature space when using Gaussians (colors) for the Fisher Vector, instead of strict borders when using k-means (dashed lines) [FS13].

particular distribution, respectively. The final feature vector  $F$  of  $ND$ -dimension for each image is then a concatenation of all feature vectors for all distributions where  $N$  is the number of distributions and  $D$  the dimension of the embedding.

### VLAD Encoding

Similar to the BOW method, a k-means with  $k$  cluster centers is used to learn a vocabulary  $\{\mu_1, \dots, \mu_k\}$ . Additionally, the distance to cluster centers is encoded, similar to the Fischer Vector. However, in contrast to the Fisher Vector, the cluster center is hard-assigned as only the distance to the closest cluster center is considered and no higher order statistics of the distribution are considered.

Every input feature  $f(x_t)$  with dimension  $D$  is assigned to its nearest cluster center  $\text{NN}(f(x_t))$ . For each cluster, all the residuals between the cluster center and the assigned features are accumulated:

$$v_i = \sum_{f(x_t): \text{NN}(f(x_t))=i} f(x_t) - \mu_i \quad (4.7)$$

The feature vector for a document can then be generated by concatenating all the  $k$  vectors  $v_i$ :

$$F = (v_1^T, \dots, v_k^T)^T \quad (4.8)$$

Thus, a document image is represented by a  $kD$ -dimensional feature vector where  $k$  is the number of clusters used for the vocabulary, and  $D$  is the dimension of the embedding.

### Whitening

Whitening of the data is applied to limit the impact of visual word co-occurrences as proposed by [JC12]. To estimate the Covariance matrix as  $\mathbf{C} = \mathbf{F} \times \mathbf{F}^T$ , the encoded features of the training database  $\mathbf{F} = [F_1 | \dots | F_n]$  are used. Each vector  $F_i$  represents the feature vector for an image in the training set after *power-law* normalization and centering around the mean. The *power-law* normalization is applied to each feature vector  $F_i = (v_1, \dots, v_{D_F})$  with dimension  $D_F$  by computing  $v_i = \sqrt{|v_i|} \cdot \text{sign}(v_i)$  for all  $1 \leq i \leq D_F$  followed by a re-normalization of  $F_i$  using the  $L_2$  norm.

Using Singular Value Decomposition (SVD) the covariance matrix  $\mathbf{C}$  is then decomposed into the diagonal matrix containing the eigenvalues  $\text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D_F}^{-\frac{1}{2}})$  and the eigenvectors  $\mathbf{V}^T$ . To reduce the dimensionality only the  $D'_F \leq D_F$  largest eigenvalues  $\lambda_i | 1 \leq i \leq D'_F$  and corresponding eigenvectors  $\mathbf{V}_{D'_F}^T$  can be kept. Whitening is then performed on the centered and *power-law* normalized feature vector  $X$  of an image as follows [JC12]:

$$\hat{X} = \frac{\text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D'_F}^{-\frac{1}{2}}) \mathbf{V}_{D'_F}^T X}{\left\| \text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D'_F}^{-\frac{1}{2}}) \mathbf{V}_{D'_F}^T X \right\|} \quad (4.9)$$

As noted by Jegou et al. [JC12] the re-normalization factor is crucial to achieving a performance improvement (they report a performance increase of up to 10% on their dataset).

Whitening is either used with one vocabulary or to jointly decorrelate multiple vocabularies. For this, multiple feature vectors with a varying number of cluster centers  $k_0, \dots, k_N$  are computed starting with a maximal number of clusters  $k_0$ , which is then halved for each following vocabulary.  $k_0$  is derived from the total number of cluster centers  $k_\Sigma$  to make the results comparable with the use of a single vocabulary:

$$k_0 = k_\Sigma \frac{1 - q}{1 - q^N} \quad (4.10)$$

$$k_n = (k_n - 1) * q \quad (4.11)$$

with  $q = 1/2$ .

### 4.3 Footwear Impressions

Similar to the retrieval of toolmarks, and handwritings, discussed in the previous sections, footwear impressions can be compared by finding matching local characteristics and comparing them using a global context. These local characteristics can either be individual characteristics or model characteristics. All shoes of the same model share the model characteristics. An example of such characteristics is, for instance, a pattern of re-occurring stars that is present on all shoe soles from the same model. However, these patterns are generally not exactly the same for all the different sizes of a specific shoe



Figure 4.11: Real crime scene footwear impression(s) collected using a gelatin foil lifter.

model, and the same pattern may be present on other shoe models as well. Individual characteristics are specific to a shoe and stem either from production or wear. Examples of this are cuts in the shoe sole from stepping onto small stones.

Therefore, like with the previous forensic domains discussed in this chapter, matching these local characteristics is essential for comparing such forensic images. However, in contrast to toolmarks, and handwriting samples discussed in this thesis, which are captured under fixed conditions, the footwear impressions discussed in this section are captured with various techniques and are therefore inherently visually distinct. Examples of such techniques are Inkless Pads, Gelatin Lifters, Casting, etc., and several different acquisition methods are discussed in Section 3.2.

Additionally, like the toolmark impressions discussed in Section 4.1, even in the case of the ‘high quality’ Inkless Pad samples, a separation of the footwear impression from the background cannot be done as easily as for the handwriting samples discussed in Section 4.2. Figure 4.11 demonstrates this in an example. The dust in the background leads to an image in which the foreground, i.e., the impression, cannot clearly be separated from the background. Furthermore, multiple impressions and blur make it hard to find one clean impression in the image. Additionally, in contrast to the toolmark impressions where it is reasonable to define areas that are the background, i.e., the lock cylinder, and areas that are the foreground, i.e., the toolmark, this distinction cannot be made for footwear impressions since, as shown in Figure 4.11, the area of the footwear impression is a blend of the impression pattern and pattern of the surface below. This issue is most pronounced in the impressions captured using Gelatin lifters, the most common form of impression seized by the Austrian Police. Like with toolmark impressions, multiple impressions can overlap. However, for footwear impressions, the whole area of the impression has to be considered due to the two-dimensional structure, and therefore overlapping impressions are more difficult to address in this case than in the case of toolmark impressions where just the edge is considered.

The annotation of the footwear impressions is more time-consuming and cannot lead to a precise identification of areas just belonging to the impression. Secondly, in contrast to toolmarks and handwritings, the proposed methodology must be robust to varying surfaces, i.e., background noise. Thirdly, the proposed similarity measure has to apply to images captured with different acquisition techniques, from ‘high-quality’ Inkless Pad impressions to impressions captured using Gelatin Lifters at crime scenes.

Therefore, the methodology proposed focuses on a trainable model that does not rely on explicitly modeling the global context. CNNs have shown a capability to model two-dimensional structures. With enough training data, the model should be able to model the local characteristics and the global context by itself and achieve a more accurate similarity measure than a complex manually designed model that incorporates both occurrences of local characteristics and their two-dimensional relative positions efficiently. Like with toolmarks, the goal is to provide forensic experts with an automated system that automatically searches through databases with tens of thousands of footwear impressions and presents the experts with the most plausible matches. Finding impressions by the same shoe requires finding individual characteristics that are hard to identify in the presence of noise. Furthermore, this would demand a time-consuming registration of impressions to compare local characteristics in areas that have already been identified as similar. Therefore, the proposed methodology does not focus on these individual characteristics but on higher-level matching structures, i.e., model characteristics. As the goal of the methodology proposed is to provide the forensic experts with a list of likely matching samples, such a detailed comparison is not essential but is an exciting topic for future work.

The methodology proposed in this section is intended to show that an approach based on metric learning based can be utilized to learn a similarity measure for footwear impression images without explicitly modeling the global context if enough training data is available. Therefore, following the comparison of metric learning approaches by Musgrave et al. [MBL20], the commonly used ResNet [HZRS16] architecture is employed in combination with the original triplet margin loss and data augmentation for rotation, translation, and scale invariance. In this section, these pre-processing techniques are explained, and subsequently, the triplet loss function and network architecture are described.

### 4.3.1 Preprocessing

Since the images are trained without explicitly extracting local characteristics, standard data augmentation techniques are employed. This pre-processing intends to aid a similarity measure that matches similar footwear impressions regardless of captured rotation, left or right shoe of a pair, the position of the impression in the images, and minor differences in scale and aspect ratio due to the acquisition process. Even though these augmentations are not intended to facilitate the matching of partial impressions, random crops should ensure that the proposed methodology works even if small parts of the impression are missing.



The pre-processing consists of three steps: first, the images are resized to a fixed width of 256 pixels. Then, a random Euclidean or affine transformation is applied, introducing random rotations, translations, and scale. In the case of the affine transformations, a random aspect ratio change is also performed within predefined bounds. Subsequently, random horizontal flips with a probability of 0.5 are utilized for left/right invariance. Finally, a random crop of size  $227 \times 227$  is taken as the input for the neural network. Since the impressions in the Impress dataset are upright, the height of the images is always greater than their width. Therefore, the resize from the first step in combination with the crop at the end assures that the random crops are overlapping but never contain the whole impression.

Even though including these augmentations does impair the similarity measure from distinguishing impressions from different shoe sizes, the increased flexibility and robustness are preferred. Additionally, since the crops are chosen to include only parts of the footwear impressions, the proposed methodology should apply to partial impressions to a certain degree. Of course, these augmentations are only employed during training; fixed center crops are used during evaluation.

### 4.3.2 Triplet Learning

As shown in Section 2.2 a significant number of metric learning losses have been developed since the classical pair-based loss was proposed by Hadsell et al. [HCL06], as, for instance, the SoftPN loss used in the methodology in Section 4.1 and Section 4.2. Nevertheless, recent investigations into these advancements by Musgrave et al. [MBL20] and Kaya et al. [KB19] show that the original triplet margin loss performs similar when paired with modern network architectures. As such the loss function is defined as:

$$\ell_{margin}(T) = \max(0, m + d(x_a, x_p) - d(x_a, x_n)) \quad (4.12)$$

The triplet  $T = \{x_a, x_p, x_m\}$  is defined with an anchor  $x_a$ , positive sample  $x_p$  and negative sample  $x_n$ . The Euclidean distance is used as a distance metric since it performs slightly better in preliminary experiments than the Cosine distance. Since the images are trained without explicitly extracting local characteristics, complex triplet selection schemes, used, for example, in Section 4.1, are not needed. Thus, a batch-based online triplet mining strategy is used to simplify the training process. In contrast to the triplet architectures used in the previous sections, the network is not explicitly split into three different branches. Instead, each batch is forwarded through the network, the triplets for the loss function are sampled from the embedded samples from a batch, and then the triplet loss function is subsequently applied to these triplets. In this way, there is no need to sample the triplets as inputs to the network explicitly, yet it has to be guaranteed that triplets can be formed in each batch, i.e., positive and negative samples can be found for each anchor. This is assured by selecting a number of  $k$  classes for each batch and drawing  $m$  samples for each class for a total number of  $n$  samples in a batch.

Additionally, to remove uninformative samples, the multi-similarity mining strategy proposed by Wang et al. [WHH<sup>+</sup>19] is employed. Similar to hard negative mining [SKP15]

this restricts the training samples to informative, i.e., hard, samples. However, it is done online during training for each batch and thus does not need separate forward passes to select informative samples ahead of each training step. The positive and negative samples are selected according to the relative similarity between the samples and an anchor  $x_a$ . In this formulation, negative pairs  $\{x_a, x_n\}$  are compared to the hardest positive pairs, i.e., the lowest similarity with maximum distance, and are only selected if the distance  $d(x_a, x_n)$  is smaller than this distance by a margin  $\epsilon$ :

$$d(x_a, x_n) < \max_{x_p \in X_P} d(x_a, x_p) + \epsilon \quad (4.13)$$

with the set  $X_P$  of all samples with the same label as the anchor  $x_a$ . Similarly, positive pairs are compared to the hardest negative pairs with the set  $X_N$  of all samples which have a different label as the anchor:

$$d(x_a, x_p) < \min_{x_n \in X_N} d(x_a, x_n) + \epsilon \quad (4.14)$$

For loss functions that do not define a fixed margin, like the multi-similarity loss proposed by [WHH<sup>+</sup>19], the parameter  $\epsilon$  directly influences which samples are considered in the loss function. Since the triplet loss function in Equation 4.12 already defines a margin  $m$  for samples that will not contribute to the loss, setting  $\epsilon \leq m$  only removes samples that are already outside the margin  $m$ . Using  $\epsilon \geq m$  restricts the samples even further.

The network architecture is split into two parts. The *trunk* with a Resnet18 [HZRS16] architecture without the last fully connected layer that has been pre-trained on ImageNet (with an error rate of 30.24<sup>2</sup>), and a single layer fully connected MLP *embedder* that maps the features computed by the *trunk* into a  $d$ -dimensional embedding. This way, the *trunk* and the *embedder* can be trained with different learning rates which is beneficial since the *trunk* is already pre-trained and the *embedder* is trained from scratch. The Resnet18 architecture with 18 layers is shown in detail in Figure 4.12. The architecture employs skip-connections, similar to the DenseNet utilized for toolmarks in Section 4.1.2. However, the architecture is frequently used and pre-trained models on ImageNet are available in commonly used frameworks like PyTorch<sup>3</sup>. Optimization is performed using Adam since it performs well even compared to more modern approaches [SSH21].

## 4.4 Summary

In this chapter, the methodology for the retrieval of forensic images was presented. The approach aims to help forensic experts retrieve the most relevant samples from a database to reduce the manual comparisons needed while searching for matching forensic samples. The methodology consists of a core trainable approach based on metric learning adapted to forensic images of three domains: toolmark analysis, writer identification and

<sup>2</sup>[https://pytorch.org/hub/pytorch\\_vision\\_resnet/](https://pytorch.org/hub/pytorch_vision_resnet/)

<sup>3</sup><https://pytorch.org/vision/main/models.html>

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 <sup>9</sup>	3.6×10 <sup>9</sup>	3.8×10 <sup>9</sup>	7.6×10 <sup>9</sup>	11.3×10 <sup>9</sup>

Figure 4.12: Comparison of different ResNet architectures [HZRS16].

retrieval, and footwear impression comparison. Metric learning enables an efficient search for similar samples in an embedding space using the Euclidean distance. Furthermore, this chapter presented a separation of the local characteristics from the global context to incorporate domain-specific constraints and utilize the available training data more efficiently.

For toolmarks, this includes four different approaches for extracting local characteristics from striated toolmark profiles or the edges of toolmark impressions. The global context is either modeled using a sliding window or a flexible DTW approach. Similarly, two methods based on SIFT keypoint locations and random patches were shown for extracting the local characteristics of handwritten documents. Since, for handwritings, the global context requires modeling the distribution of these local characteristics, VLAD and Fisher Vector were proposed to encode this distribution. In contrast, for footwear impressions, an end-to-end-based methodology was presented, which does not require explicit modeling of the global context nor explicit extraction of the local characteristics. The approach shown does, however, use data augmentation techniques to utilize the available data more efficiently. The proposed core metric learning methodology utilizes either shallow CNN architectures (similar to the PN-Net used as a local descriptor) or deeper ResNet-like architectures based on skip connections. All methods presented use triplet-based loss functions to learn the relationship between the training samples.



# Evaluation

The methodology for comparing forensic images proposed is evaluated in this section. The main goal of this chapter is to investigate if the proposed metric learning-based approaches can be used to support the work of forensic experts. For this, the datasets presented in Chapter 3 and publicly available datasets for each of the three forensic images domains discussed, namely toolmarks, footwear impressions, and handwritings, are utilized. As the use case for these methodologies is the retrieval of forensic images by forensic experts, the evaluation in this chapter is approached in terms of information retrieval. As such the problem is formulated as follows: a user expresses an *information need* using a set of queries and retrieves *relevant* and *non-relevant* documents from a *document collection* [MRS08]. For example, in the case of toolmarks, the *information need* can be expressed as the search for a similar toolmark, i.e., the search for a toolmark made by the same tool. This way, *relevant* and *non-relevant* documents are represented by toolmarks made by the same or another tool, respectively, and this representation works analogously for footwear impressions made by the same shoe and handwritings made by the same author. The quality of the retrieved documents is defined by how many relevant documents are retrieved. Ideally, the first results retrieved are the relevant documents in the document collection for each information need.

Crucially, the retrieval is, if not stated otherwise, defined as an open-set problem with distinct sets of labels for the training set and testing set. Open-set means the training set is only used to learn valuable features to identify similar and dissimilar samples, but it is not used to learn how to directly identify the labels of the samples as in classification tasks. Requiring all labels to appear in the training set would pose several problems for forensic images: firstly, forensic experts would need to maintain a meticulously labeled collection in which for each sample added to the collection, a label is identified. This labeling entails comparing each new sample either manually or semi-automatically to all the samples currently present in the collection and identifying if it represents a new label or if the label is already present. Secondly, it would require re-training the system each

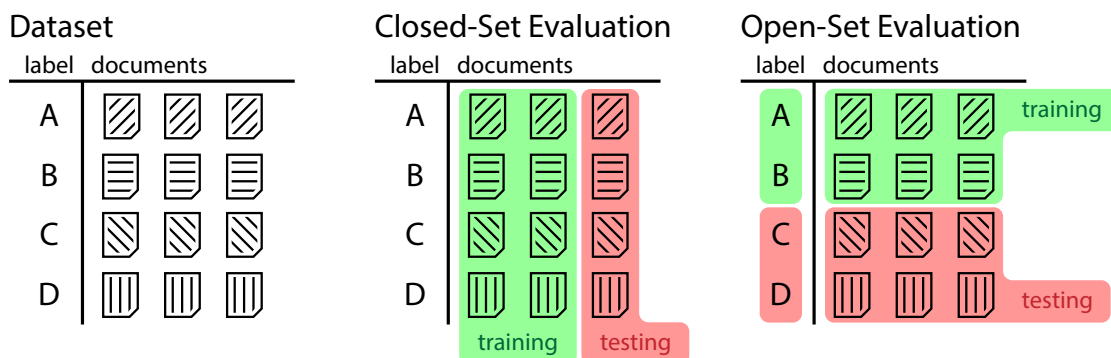


Figure 5.1: Visualization of the difference between open- and closed-set evaluation.

time a new label is identified, and since there might only be one or a few samples per label for forensic collections, new labels would be identified regularly. Instead, using an open-set problem definition allows an evaluation of the performance, focused on assisting forensic experts without additionally increasing their workload. The difference between performing a closed-set and open-set evaluation is shown in an example in Figure 5.1. In a closed-set evaluation, all the labels are used in both the training and testing sets. However, the documents are split between the testing and training sets individually for each label. Contrary to this, the labels are divided into training and testing sets in an open-set evaluation. All the corresponding documents for each label are then put either in the training set or the testing set. Therefore, for a closed-set evaluation, each label must contain at least two different documents in the dataset to work with such an evaluation scheme which may not be the case for forensic image datasets.

In this chapter, first, the different evaluation metrics used are described in detail in Section 5.1. The general idea is to use metrics that faithfully capture the quality of the retrieved results and metrics that can be intuitively explained to forensic experts. Nevertheless, since the provided results must be comparable to other publications, standard metrics used in the respective fields are used when the results are related to the state of the art. In the subsequent sections, the results are presented and discussed in detail for each forensic domain discussed in this work, i.e., striated toolmarks, toolmarks impressions, handwritings, and footwear impressions.

## 5.1 Metrics

Generally speaking, to assess the effectiveness of an information retrieval system, it is of interest how many of the relevant documents in the collection are returned for a query and what fraction of the returned results are relevant; which can be quantified with recall and precision, respectively [MRS08]:

$$\textit{precision} = \frac{\text{number of relevant items retrieved}}{\text{number of items retrieved}} \quad (5.1)$$

$$recall = \frac{\text{number of relevant items retrieved}}{\text{number of relative items}} \quad (5.2)$$

Since these metrics present a trade-off, the F measure, or F-Score, can be used as a single value to capture this relationship. The balanced  $F_1$  measure that weights recall and precision evenly is defined as [MRS08]:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (5.3)$$

These measures are set based, and as such, they are based on a set of a fixed number  $k$  of retrieved documents that contains relevant and non-relevant items. In the forensic context, the  $F_1$  score allows an assessment of how well a system can distinguish known matches from known non-matches, i.e., determine if two DNA samples or fingerprints match, and is similarly used in Section 5.2 for striated toolmarks to compare to the state of the art.

However, such systems are designed with a fixed threshold and work fully automated without incorporating forensic experts. In contrast to this, the methods proposed in this thesis are not designed to give the forensics experts a binary “yes” or “no” answer, but instead a ranked list of results sorted by relevance. In order to evaluate such a ranked retrieval, the measures presented above have to be extended. By varying the number  $k$  of retrieved documents and calculating precision and recall for each  $k$ , the ranked retrieval performance can be quantified [MRS08]. The result is commonly plotted with the precision on the y-axis and the recall on the x-axis [MRS08]. However, this graph only shows the system’s effectiveness for a single query, i.e., information need. The arithmetic mean of precision can be used to calculate such plots for a whole set of queries, which is how the precision/recall plots in this thesis are generated.

Even though these precision/recall plots provide an informative assessment of the retrieval performance, in order to compare results, a single number is often preferred. A simple solution for this is to take a fixed  $k$  and calculate the *precision at  $k$*  [MRS08]. Similarly, for local image similarities, a commonly used measure is the False Positive Rate at 95% recall (FPR@95), which is, for instance, used in Section 5.3 to compare to the state of the art; a definition can be found in [MRS08]. The FPR95 allows an intuitive assessment of the expected false positives for a system that correctly identifies nearly all the true positives, i.e., 95%. A measure to capture the quality across the whole precision/recall curve is the Mean Average Precision (MAP) which is calculated as follows [MRS08]:

$$MAP(Q) = \frac{1}{|Q|} \cdot \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} precision(R_{jk}) \quad (5.4)$$

with *information needs*  $q_j \in Q$ , the *set of all information needs*  $Q$ , *relevant documents*  $\{d_1, \dots, d_{m_j}\}$ , and  $R_{jk}$  the minimal set of ranked retrieval results containing  $d_k$  [MRS08]. In the example of toolmark images,  $q_j$  can be formulated as “find images with toolmarks

made by the same tool as the supplied image” and  $d_k$  as “image with toolmarks made by the same tool”. The retrieval results for each  $q_i$  are ranked by similarity score. Each  $R_{jk}$  then contains  $d_k$  and all other images which are more similar to the supplied image than  $d_k$ . A perfect score of 1.0 is achieved when all  $d_k$  are ranked at the top, and thus all  $R_{jk}$  contain only relevant documents.

Even though the MAP is widely used and helps compare different methodologies [MRS08], e.g., to the state of the art, it does not provide an intuitive understanding of the ranking performance that forensic experts request. Such a measure that can be easily interpreted is the top-k hard criterion, which is, for instance, commonly used to evaluate the performance of writer retrieval methods [FKD<sup>+</sup>17], is defined as follows:

$$\text{hard-}k = \frac{\sum_{\text{query}} k \text{ relevant items found in top-}k \text{ results}}{\text{number of queries}} \quad (5.5)$$

It simply states the probability that all retrieved documents are relevant if  $k$  documents are retrieved. However, since this probability depends on the total number of relevant documents in the document collection, it is only meaningful to compare the relative performance of different methods on the same dataset. Similarly, the measure may not be practical with unbalanced datasets since it does not capture how many of the relevant documents are not retrieved. Therefore, multiple top-k scores have to be combined for an in-depth assessment of the performance as some relevant items might be inherently easier to retrieve than others. As a side note, the hard top-1 criterion is similar to the accuracy computed when evaluating neural networks on classification tasks in which also just the top-ranked result, i.e., the class with the highest log-likelihood, is taken as the predicted label.

A similar criterion that is also used in the evaluations of writer retrieval publications [FKD<sup>+</sup>17] is the top-k soft criterion which, in contrast to the hard criterion, captures the probability that at least one of the retrieved documents is relevant:

$$\text{soft-}k = \frac{\sum_{\text{query}} \text{relevant item found in top-}k \text{ results}}{\text{number of queries}} \quad (5.6)$$

The soft-k criterion provides an intuitive measure of how likely it is to find any relevant item when manually inspecting the first  $k$  ranked documents. Since it might be enough for forensic experts to find any matching item in the collection to start an investigation, this is a reasonable simplification in the case of forensic images. However, since a number  $k$  has to be selected, it provides only limited expressiveness, similarly to the top-k hard criterion. Nevertheless, by varying  $k$ , the Cumulative Match Characteristic (CMC) can be plotted, which is frequently used by publications in the field of footwear impression retrieval like Kong et al. [KSRF19] and Kortylewski et al. [KAV15]. In the CMC, the Cumulative Match Score (CMS), i.e., top-k soft criterion, is placed on the y-axis, and the number of samples retrieved  $k$  is placed on the x-axis as a fraction of the whole collection. It enables an intuitive understanding of how many samples must be retrieved to find a relevant document with a certain probability. Moreover, it shows how probable it is



to find a relevant document when a certain amount of documents is retrieved from the collection.

In order to evaluate a retrieval system and calculate the metrics described above, for each dataset which consists of multiple images, the queries, document collection, and relevant and non-relevant documents for each query have to be selected. In this thesis, the evaluation is done using a leave-one-out strategy. Each item in the dataset is taken once as the query, i.e., information need. All other items in the dataset define the retrievable document collection for each query. The actual retrieval is done by computing the similarity between the query and each document in this collection. The results are then subsequently ranked by this measure. The relevance of a retrieved document is defined by its label and the label of the query. If these labels match, the item is relevant; otherwise, it is not relevant. The set of all information needs are all the images in the dataset, and for each information need, the set of relevant documents is all the matching images in the dataset except the current query image. Depending on the distance measure  $d$ , up to  $N \times N$  distance calculations must be performed for the evaluation. For symmetric distance measures, this can be sped up by only computing  $d(A, B)$  and not  $d(B, A)$ .

## 5.2 Striated Toolmarks

In this section, the methodology proposed in Section 4.1 is evaluated using striated toolmarks. The evaluation is based on the NFI database of 300 striated screwdriver toolmarks published by Baiker et al. [BKP<sup>+</sup>14] presented at the beginning of this section. Other works on that data like [BKP<sup>+</sup>14, BJJK10] focus on how well an algorithm can identify pairs of known matches or pairs of known non-matches and are based on approaches that do not utilize machine learning and thus do not require the distinct separation of training and testing data. Consequently, the published results can not directly be compared to the deep-learning-based TripNet methodology proposed. With this in mind, a curvature matching method is proposed as a baseline approach. This approach achieves a similar performance as the methodology proposed by Baiker et al. [BKP<sup>+</sup>14]. It does not require training data and can thus faithfully be compared with state-of-the-art methods. Subsequently, the performance improvements achieved by the TripNet methodology are investigated using this baseline. This comparison is made using the information retrieval metrics described. Two different partitionings of the NFI database, namely NFIT and SPLIT, are created to enable both a close-set and open-set evaluation with separate training and testing sets. The impact of decoupling local characteristics is investigated utilizing permuted profiles and profile segments compared to modeling the global context using full profiles. Finally, the results are discussed in detail, and the advantages and disadvantages of the proposed methodology are presented.

### 5.2.1 Dataset

The NFI dataset published by Baiker et al. [BKP<sup>+</sup>14] consists of 300 toolmarks from 50 different tools. For each tool, toolmarks were made with different angles of attack ( $\alpha$ ),

i.e.,  $\alpha = 15^\circ, 30^\circ, 45^\circ, 60^\circ,$  and  $75^\circ$  are available. For 10 tools additional 5 toolmarks each at  $\alpha = 45^\circ$  are provided. However, since a balanced dataset is preferred, these additional  $45^\circ$  toolmarks are ignored in the NFIT and SPLIT partitionings described below. All toolmarks are available as 2D images, 3D surfaces, or preprocessed 1D profiles extracted from the surfaces. For evaluating the baseline and TripNet, the profiles and the 2D images are used, respectively.

Since the 2D images are not preprocessed, as opposed to the 1D profiles in the NFI dataset, a rough manual alignment using translation, scale, and rotation is performed by hand, as shown in an example in Figure 5.2. Further, to increase the number of samples for training and testing TripNet, vertically flipped (reflected along the central horizontal axis) versions are added to the set of 2D images. Since the position of local characteristics is a defining factor for a tool, these images are assigned a distinct set of an additional 50 tools. This augmentation artificially doubles the number of images to 400 and the number of tools to 100.

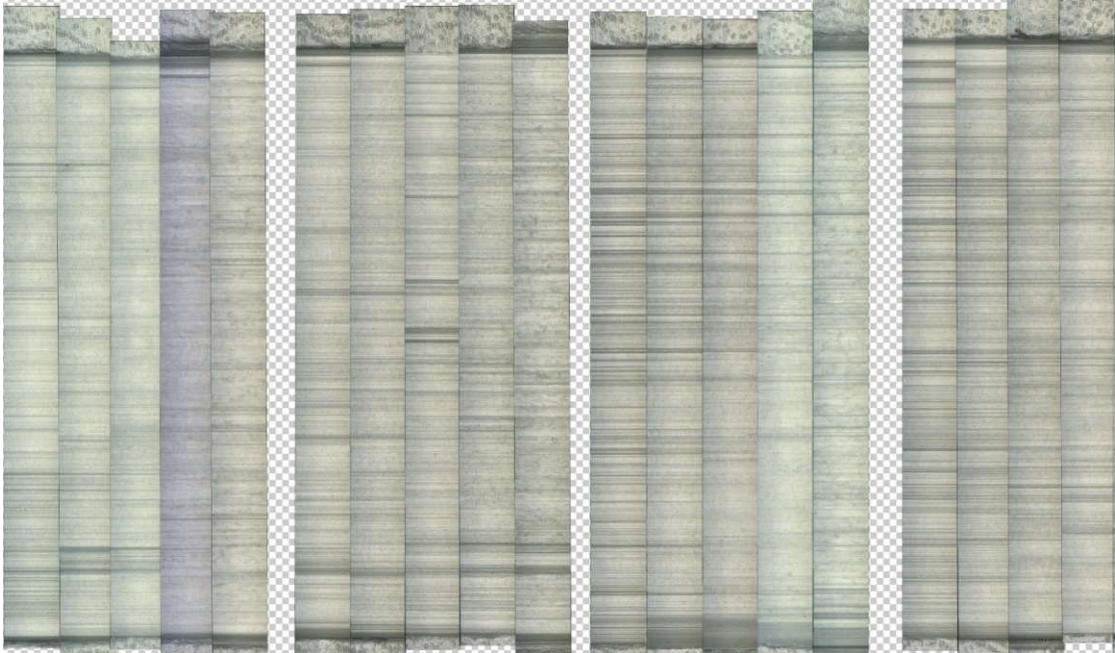


Figure 5.2: Manual alignment process of the 2D images in the NFIT dataset shown on an example.

In order to allow a comparison with [BKP<sup>+</sup>14] the KM (Known Match) 15 vs. KNM (Known Non-Match) and KM 15/30 vs. KNM partitionings presented there are evaluated. These include all comparisons between all matching toolmarks with a difference in  $\alpha$  of  $15^\circ$ , and  $15^\circ$  or  $30^\circ$  with all non-matching distances of  $\alpha = 45^\circ$ , respectively. The additional  $\alpha = 45^\circ$  toolmarks are not included in these sets.

For the training-based TripNet, two different partitioning of the NFI dataset, NFIT

and SPLIT, are proposed to perform a closed-set and open-set evaluation, respectively. The NFIT dataset is partitioned into training and testing: all toolmarks of a particular  $\alpha$  (including their flipped counterparts for TripNet) are put into the test set; all other toolmarks into the training set. The naming of the partitioning reflects the toolmarks in the test set, e.g., NFIT 15 contains all toolmarks with  $\alpha = 15^\circ$  in the test set.

These NFIT partitionings allow assessing the performance of the method proposed for finding a matching tool in an annotated and trained database. However, this does not capture use-cases where retraining the CNN for new tools is not desired or feasible. Therefore, the whole dataset is additionally split into toolmark images from tools with even and odd numbers, resulting in a training set with 24 tools and a testing set with 26 tools. The additional  $\alpha = 45^\circ$  images are omitted. The testing set is partitioned similarly as above into the SPLIT 15, SPLIT 30, SPLIT 45, SPLIT 60, and SPLIT 75 datasets.

### 5.2.2 Elastic Shape Matching Baseline

The baseline is based on the elastic shape metric proposed by Srivastava et al. [SKJJ11]. This approach is publicly available<sup>1</sup> and requires no parameter evaluation. For comparing shapes of closed and open curves in  $\mathbb{R}^n$ , the distance is expressed as a combination of bending and stretching deformations. In contrast to other elastic shape metrics, the curve is represented by the Square-Root-Velocity (SRV) function to reduce it to an  $L^2$  metric. All curves are scaled to unit length in order to achieve scale invariance. These open curves with unit lengths are then represented by points on a unit hypersphere in this *pre-shape-space*  $L^2(D, \mathbb{R}^n)$ . The distance between two curves is then defined by the length of the minimizing geodesic between their point representations in *pre-shape-space*. Since this *pre-shape-space* is not invariant to rotation and re-parameterization an additional optimization step is performed afterwards to compute the distances in *shape-space*. The methodology is described in detail in [SKJJ11].

This approach is directly applied to the NFI Toolmark profiles after downsampling to 800 points, which corresponds to the minimal wavelength used by Baiker et al. [BKP<sup>+</sup>14]. The extensive preprocessing pipeline applied to the NFI profiles is described in [BKP<sup>+</sup>14] and includes cropping, stitching, alignment, global shape removal, and noise reduction.

In Table 5.1 the baseline is compared to the results published by Baiker et al. [BKP<sup>+</sup>14]. In order to allow a one-score comparison, the False Discovery Rate (FDR) and Negative Predictive Value (NPV) given by Baiker et al. are converted into  $F_1$  scores. The two methods perform the same for the KM 15 vs. KNM evaluation. However, for the dataset containing both  $15^\circ$  and  $30^\circ$  comparisons, the baseline performs slightly worse with an  $F_1$  score of 0.75 compared to 0.79 achieved by the method proposed by Baiker [BKP<sup>+</sup>14]. This difference suggests that the baseline is not suited as well for  $\alpha = 30^\circ$ . Still, the general trend that comparisons of samples with  $\alpha = 15^\circ$  work well, but the performance decreases drastically for  $\alpha > 15$  can be observed for both approaches. The performance decline of the baseline with increasing  $\alpha$  difference is also shown in Figure 5.3. All

<sup>1</sup><http://ssamg.stat.fsu.edu/software>

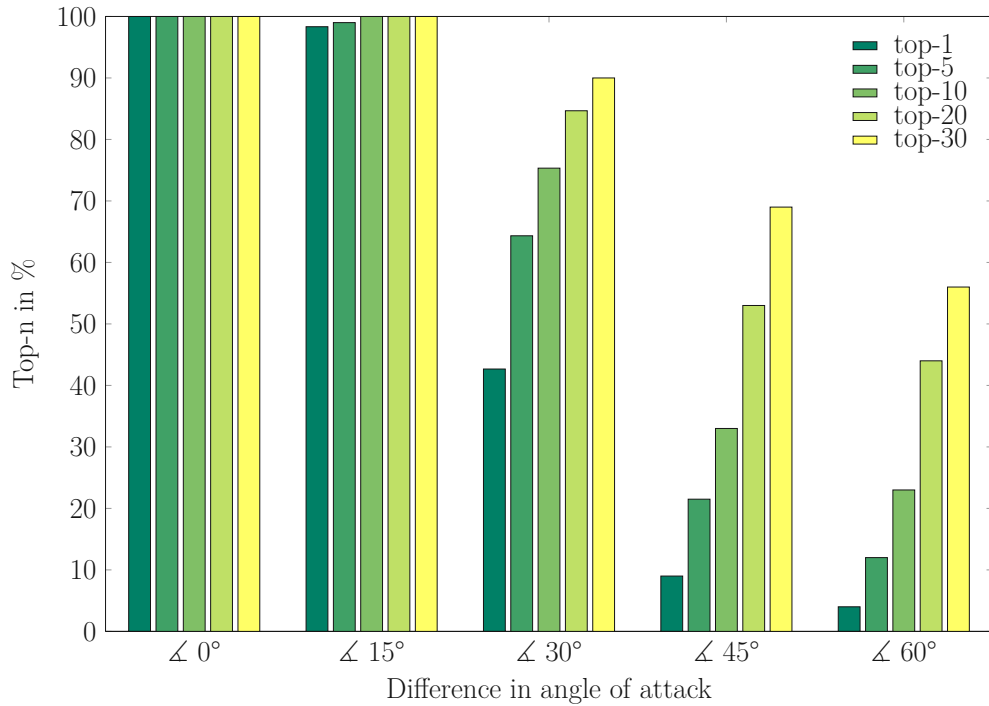


Figure 5.3: Comparison of the soft-criteria scores achieved by the baseline approach for  $\alpha$  differences of 0° to 60°.

comparisons were restricted to the given  $\alpha$  difference for this experiment. In order to allow a comparison with  $\alpha$  differences of 0° the additional  $\alpha = 45^\circ$  toolmarks were included; all images from the NFI toolmark database were used. However, these additional  $\alpha = 45^\circ$  toolmarks are only available for a subset of just ten different tools, and as such, the perfect scores achieved here are of limited relevance.

	Metric	Baiker [BKP <sup>+</sup> 14]	Baseline	TripNet Full Profile
KM 15 vs. KNM	$F_1$	<b>0.96</b>	<b>0.96</b>	
KM 15/30 vs. KNM	$F_1$	<b>0.79</b>	0.75	
NFIT 15	MAP		0.47	<b>0.78</b>
NFIT 30	MAP		0.69	<b>0.95</b>
NFIT 45	MAP		0.70	<b>0.94</b>
NFIT 60	MAP		0.56	<b>0.84</b>
NFIT 75	MAP		0.35	<b>0.54</b>

Table 5.1: Results for the baseline and TripNet in comparison with [BKP<sup>+</sup>14].

For the NFIT datasets, the results are similarly dependent on the  $\alpha$  difference of the compared toolmarks. In the NFIT 45 dataset, which contains just comparisons with  $\alpha$  differences of  $15^\circ$  and  $30^\circ$ , a MAP of 0.70 is achieved. With increasing  $\alpha$  difference the MAP drops to 0.47 for NFIT 15 and even 0.35 for NFIT 75 which both contain  $\alpha$  differences of  $15^\circ$  to  $60^\circ$ . In Figure 5.5 the steep decline for a recall greater than 0.3 suggests that after correctly identifying the samples with similar  $\alpha$ , the approach fails to distinguish the remaining toolmarks.

Since this approach computes similarities without prior training, similar performance is achieved on the SPLIT datasets as shown in Table 5.3. The slight improvements in MAP are because only tools with odd numbers are in the testing set, and therefore the total number of relevant and non-relevant images is reduced.

### 5.2.3 TripNet

In this section, the TripNet CNN described in Section 4.1.2 with the SoftPN loss [BPG<sup>+</sup>16] is evaluated on striated toolmark images. Since the proposed methodology requires separate training and testing sets, the evaluation is performed on the NFIT and SPLIT partitionings described above. Similar to the processed 1D profiles used for the baseline, the 2D images are uniformly downscaled to a height of 800 pixels. The resulting resolution is about 100 pixels per millimeter. The triplet creation is done online, i.e., not created beforehand but during training. Min/max-normalization and mean pixel subtraction are performed as a preprocessing step.

A distance calculation between two toolmark profiles takes about 3s on an Intel i7-5500U CPU for the baseline. Since  $N \times N$  (9,000) computations are required, it takes 25h to calculate all distances for the whole NFI Toolmark dataset. In contrast, the embedding calculation for TripNet is done in 0.01ms once the toolmark images are in memory; otherwise, it takes 1ms. All experiments for TripNet were performed using an NVIDIA Titan X (Maxwell architecture).

For the extraction of local characteristics, three different strategies are evaluated, namely full profiles, permuted profiles, profile segments, and patches, as defined in Section 4.1.1. The similarity is computed for the full profiles and permuted profiles by calculating the  $L_2$  distance of the embedding vectors returned by the TripNet. As uncoupling the local characteristics requires explicitly modeling the global context, for the profile segments and patches a simple slinging window, as explained in Section 4.1.3, is utilized. Training of the TripNet is done using Stochastic Gradient Descent with a learning rate of 0.01, weight decay of  $10^{-4}$ , and momentum of 0.9.

This section first presents the results achieved using full profiles on the different NFIT partitionings described above. It is shown in detail that the proposed methodology performs significantly better than the baseline, especially for a bigger  $\alpha$  difference. Additionally, the impact of an  $\alpha$  difference on the TripNet performance is explored. Furthermore, the influence of the embedding dimension is investigated. Subsequently, an open-set evaluation is performed using the SPLIT partitionings using full profiles,

permuted profiles, profile segments, and patches, demonstrating that uncoupling the local characteristics improves the performance considerably on striated toolmarks from unseen tools.

## NFIT

The results attained on the NFIT datasets indicate that the basic TripNet with full profiles is better suited to handle  $\alpha$  differences bigger than  $15^\circ$  than the baseline. The resulting MAP of over 0.9 shown in Table 5.1 suggests that most of the matching toolmarks are ranked at the top for NFIT 45 and NFIT 30. For NFIT 15 and NFIT 60 the MAP declines slightly to 0.78 and 0.84. However, for NFIT 75, a MAP of only 0.54 is achieved, even though the distribution of  $\alpha$  differences is identical to NFIT 15. Degradation of the toolmarks for greater  $\alpha$  can explain this, which is also suggested in [BKP<sup>+</sup>14] and indicated in Table 5.2. The same can be observed when comparing the result of NFIT 30 with NFIT 60. Table 5.2 filters the results on each partitioning by  $\alpha$ -differences to isolate the  $\alpha$ -difference of the retrieved results. This separation demonstrates that overall the retrieval of toolmarks is more challenging for NFIT 15, NFIT 60, and NFIT 75 compared to NFIT 30 and NFIT 45, even for an alpha difference of only  $15^\circ$ . Nevertheless, even for NFIT 15, a MAP of 0.79 can be achieved when only toolmark images with an  $\alpha$  of  $75^\circ$  are considered for retrieval, i.e., an  $\alpha$ -difference of  $60^\circ$ .

$\alpha$ -difference	$-15^\circ$	$+15^\circ$	$-30^\circ$	$+30^\circ$	$-45^\circ$	$+45^\circ$	$-60^\circ$	$+60^\circ$
NFIT 15		0.88		0.82		0.81		0.79
NFIT 30	<b>0.99</b>	<b>0.97</b>		0.95		<b>0.94</b>		
NFIT 45	0.98	0.96	<b>0.98</b>	<b>0.96</b>				
NFIT 60	0.92	0.88	0.86		<b>0.85</b>			
NFIT 75	0.69		0.58		0.57		0.61	

Table 5.2: Results in MAP achieved by the TripNet with full profiles on the NFIT datasets filtered by  $\alpha$ -difference.

In Figure 5.4 the retrieval results are shown in detail as Precision/Recall plots emphasizing that the method proposed works well for NFIT 30 and NFIT 45, not at all for NFIT 75, and somewhere in between for NFIT 15 and NFIT 60.

Precision/Recall plots are compared in Figure 5.5 to investigate the impact of the embedding dimension for NFIT 15. In the case of a dimension of 16, the network performs worse than the baseline. The sharp drop at a recall of 0.05 suggests the network has problems in distinguishing all toolmarks with an  $\alpha$  difference of  $15^\circ$ . Nevertheless, the baseline approach is outperformed by all networks with an embedding dimension of 32 or greater. Increasing the embedding dimension to more than 64 does not further improve results.

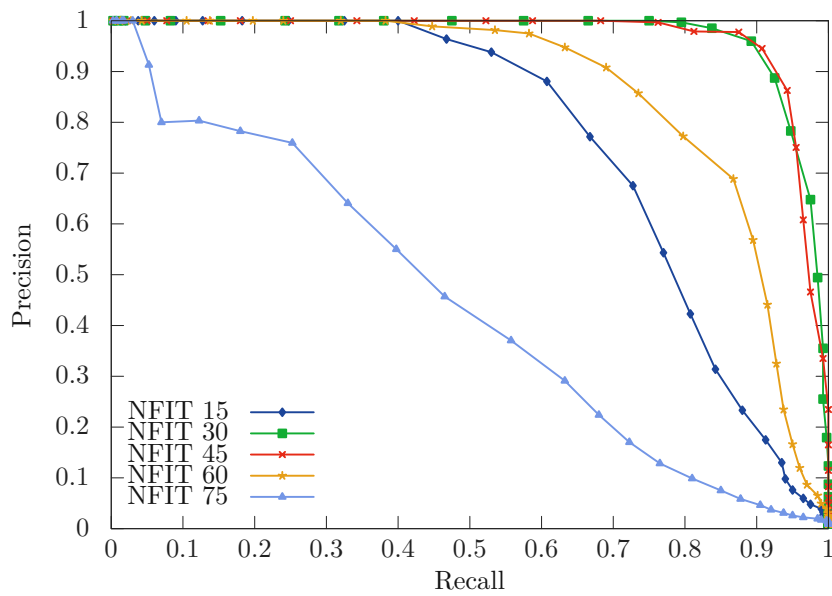


Figure 5.4: Precision/Recall plot for TripNet with full profiles comparing different partitionings of the NFI Toolmark dataset.

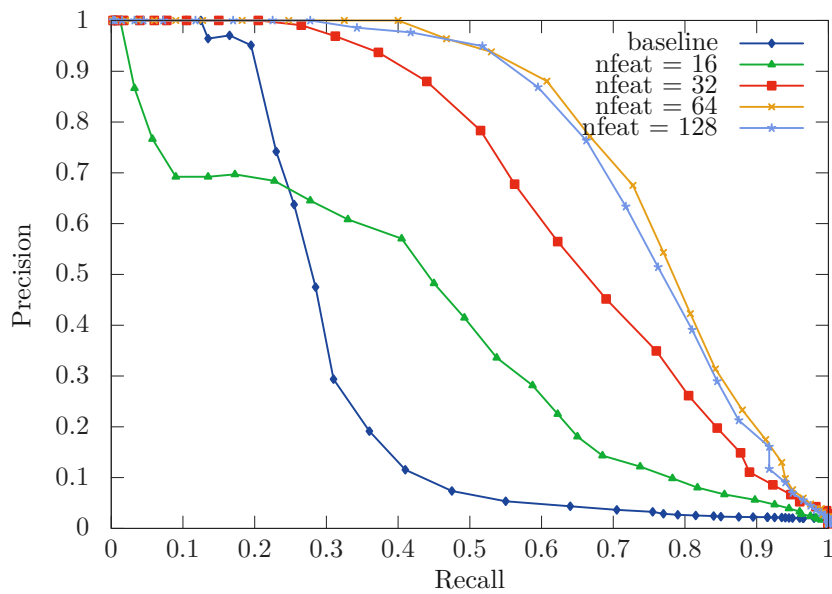


Figure 5.5: Precision/Recall plot for TripNet with full profiles comparing different embedding dimensions using NFI 15.

## SPLIT

Proceeding with the SPLIT datasets, which allow an open-set evaluation of the performance for striated toolmark from unseen tools, the TripNet with full profiles performs significantly worse than the baseline. Table 5.3 shows MAP of only 0.27 is achieved for SPLIT 15 compared to 0.55 for the baseline. Even though the approach can successfully separate toolmark images from different tools, it cannot generalize well to unseen tools not present in the training set.

	Metric	Baseline	TripNet Full Profile	TripNet Segments
SPLIT 15	MAP	0.55	0.27	<b>0.56</b>
SPLIT 30	MAP	<b>0.75</b>	0.35	<b>0.75</b>
SPLIT 45	MAP	0.75	0.31	<b>0.77</b>
SPLIT 60	MAP	0.61	0.23	<b>0.72</b>
SPLIT 75	MAP	0.41	0.16	<b>0.44</b>

Table 5.3: Results for the retrieval of toolmark images of unseen tools.

Figure 5.6 shows a Precision/Recall plot in which the different strategies for uncoupling the local characteristics proposed in Section 4.1 are compared using the SPLIT 15 dataset. Since the available number of samples is relatively small, introducing random permutations to the profiles in order to extrapolate the training data already improves the MAP from 0.27 to 0.36. As expected, the increased MAP of 9% shows that decoupling the local characteristics of the toolmarks from their position during training is advantageous since this artificially increases the training set, and the CNN learns that not only the presence but also the position of a local characteristic is essential. However, it still performs worse than the baseline. Using randomly extracted  $1 \times 48$  pixel ( $\approx 480 \mu\text{m}$ ) segments, and thus decoupling the local characteristics from the position on the profile completely during training, leads to further improvements with a MAP of 0.56. Even though the MAP is only slightly better than the baseline, the detailed Recall/Precision plots show that the resulting similarity measure is still more distinctive. The results filtered by  $\alpha$ -difference in Table 5.4 indicate that for unseen tools, this method only works well for differences of  $15^\circ$  with MAPs between 0.81 and 1.00. In case toolmarks with an  $\alpha$  of  $75^\circ$  are not considered, MAPs of at least 0.66 are achieved for  $\alpha$ -differences of  $30^\circ$ . Nevertheless, comparisons including toolmarks with an  $\alpha$  of  $75^\circ$  perform significantly worse than all others, which is also observed in the previous section with the NFIT 75 dataset. Using patches instead of segments in order to improve robustness leads to similar performance but does not offer measurable improvement in MAP, and the Precision/Recall plot is slightly worse than with segments, as shown in Figure 5.6.

In Figure 5.7 the impact of the segment size on the performance is depicted using the SPLIT 15 dataset. Overall, the performance is similar, with a MAP ranging from 0.52 to 0.56. The best performance is achieved with  $1 \times 48$  pixel segments. Table 5.3 shows that



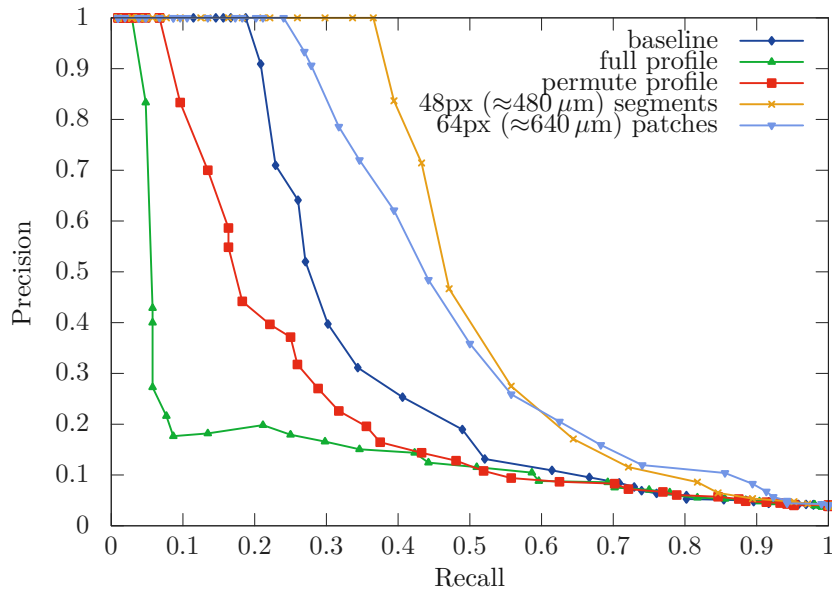


Figure 5.6: Precision/Recall plot comparing the proposed TripNet approaches using the SPLIT 15 dataset.

$\alpha$ -difference	$-15^\circ$	$+15^\circ$	$-30^\circ$	$+30^\circ$	$-45^\circ$	$+45^\circ$	$-60^\circ$	$+60^\circ$
SPLIT 15		<b>1.00</b>		0.66		<b>0.45</b>		0.35
SPLIT 30	0.98	<b>1.00</b>		<b>0.71</b>		0.38		
SPLIT 45	<b>1.00</b>	0.98	0.68	0.55				
SPLIT 60	0.98	0.81	<b>0.69</b>		<b>0.48</b>			
SPLIT 75	0.82		0.57		0.32		0.35	

Table 5.4: Results in MAP achieved by the TripNet with Segments on the SPLIT datasets filtered by  $\alpha$ -difference.

this approach, using segments with  $1 \times 48$  pixel, achieves at least the same performance as the baseline and can lead to a performance increase of up to 11% MAP depending on the dataset. Even though the results on the SPLIT datasets are not as promising as on the NFIT datasets, an overall MAP of over 0.70 can be achieved for SPLIT 30, SPLIT 45, and SPLIT 60, which only have a maximum  $\alpha$  between the query and the search images  $\alpha$ -differences of  $15^\circ$  to  $45^\circ$ . In Figure 5.8 the Precision/Recall plots for the different datasets are compared in detail.

#### 5.2.4 Discussion

As shown, a primary challenge for matching striated toolmarks is to handle differences in angle of attack. This section evaluates two approaches, an elastic shape matching

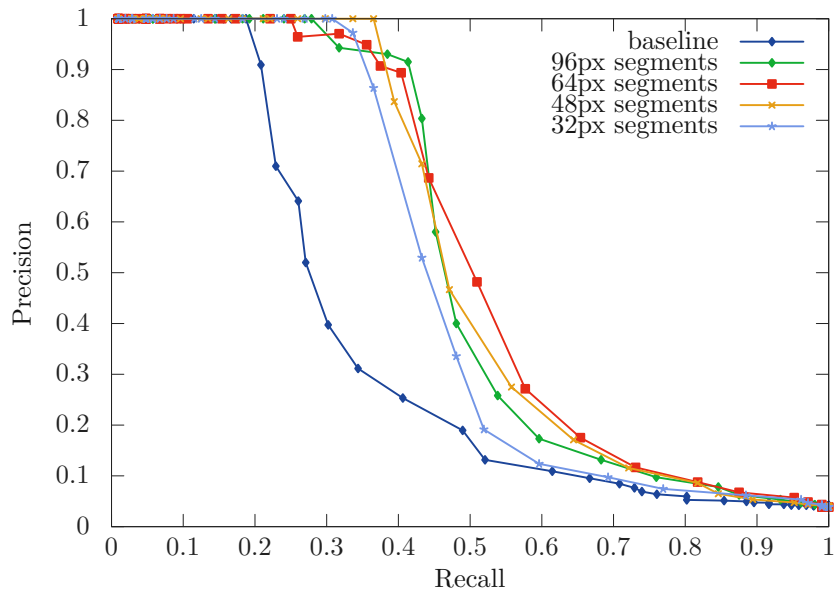


Figure 5.7: Precision/Recall plot comparing the effect of varying segment length using the SPLIT 15 dataset.

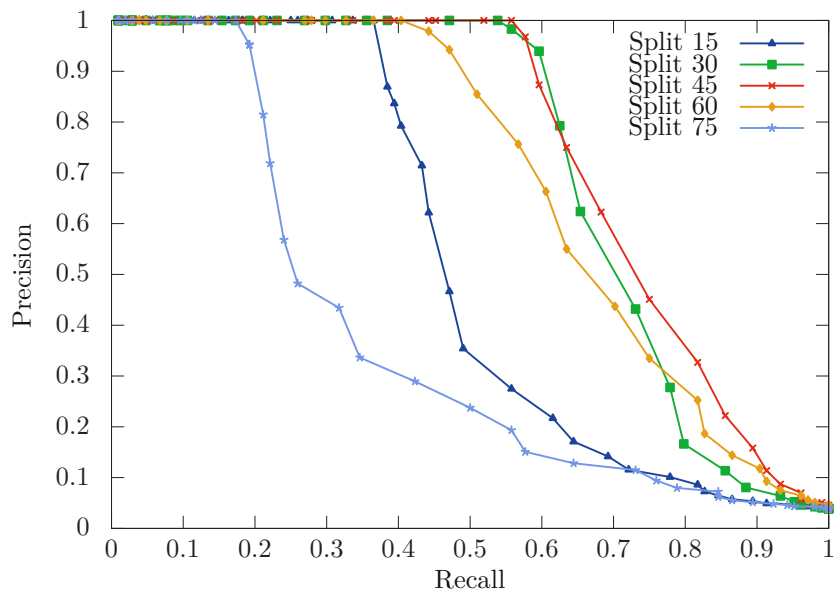


Figure 5.8: Precision/Recall plot comparing the performance of the TripNet with  $1 \times 48$  segments on the SPLIT datasets.

baseline, and a neural network-based TripNet. Even though a perfect score is achieved by the baseline when comparing toolmarks made with the same  $\alpha$ , it is not suited for differences of more than  $15^\circ$ . It can be seen in Table 5.1 by the performance drop from an  $F_1$ -Score of 0.96 to 0.75 from KM 15 vs. KNM to KM 15/30 vs. KNM that the baseline approach does not work well for distinguishing toolmarks with an  $\alpha$  difference of more than  $15^\circ$ . Further, due to the high computational demands of about 3s per comparison, this approach is restricted to small toolmark databases in environments without time constraints.

The TripNet handles these situations better. Even though the NFI Toolmark dataset is fairly small, the performance achieved by TripNet is promising. As demonstrated, the network can adapt to  $\alpha$  differences of  $15^\circ$  to  $60^\circ$ , achieving a MAP of 0.78 for the NFIT 15 partitioning. For  $\alpha$  differences of  $15^\circ$  to  $45^\circ$  in the NFIT 30 partitioning, a MAP of 0.95 is achieved. Still, there is room for improvement, especially for the most challenging NFIT 75 dataset. However, the results are still unsatisfactory for extreme cases like NFIT 75.

Even though the performance of the TripNet with full profiles for toolmarks of unseen tools cannot compete with the above results, by using profile segments instead, a MAP of 0.75 can be achieved on the SPLIT 30 dataset with  $\alpha$  differences of  $15^\circ$  to  $45^\circ$ . Furthermore, the baseline approach is outperformed by up to 9% MAP on this task, even though the calculations are significantly faster; i.e., the computation of all distances in the SPLIT 15 dataset takes about 20s instead of several hours. However, when comparing Table 5.1 with Table 5.3 the performance of TripNet for toolmarks of unseen tools still leaves room for improvement. Since the dataset is relatively small, the proposed uncoupling of the local characteristics from the global context by dividing the profiles into segments improves the performance considerably. However, this uncoupling introduces a handcrafted sliding window approach for combining the segment representations in the embedding, which is not advanced enough to capture additional higher-level information encoded in the profiles. Additionally, manual translation, rotation, and scale correction are essential for the current network and dataset since the performance degrades significantly to a MAP of just 0.42 for NFIT 15. This drop in performance does not occur when the preprocessed 1D profiles are used, although this significantly impairs the network since no random crops can be extracted during training. Therefore, the ability of the network to adapt to variations in the data is severely limited. In this case, the MAP drops from 0.78 to 0.67.

### 5.3 Toolmark Impressions

The previous section evaluated the methodology proposed on striated toolmarks created under laboratory conditions. It demonstrated that, especially for the most challenging dataset partitionings, the decoupling of local characteristics, as proposed in Section 4.1, can improve the performance significantly. In this section, the methodology proposed in Section 4.1 is evaluated on impression toolmarks from real criminal cases. In contrast to the striated toolmarks discussed in the previous section, the impression images contain

toolmark impressions and the structure of the cylinder locks. Separating the toolmarks in these images from the background is a challenging task that is not the focus of this thesis. Therefore, the FORMS dataset contains manual annotation for the edges of the toolmark impressions. The FORMS dataset used for the evaluation in this section is discussed in detail in Section 3.1. Since impression marks can more reliably be identified than striation marks in these kinds of images, only impression marks are considered in this dataset.

As shown in the previous section, the uncoupling of local characteristics can be beneficial, and this section investigates the impact of increasing the complexity of the modeled global context. Even though striated toolmarks replicate the same toolmark edge as in the toolmark impression, modeling it as a sequence of two-dimensional local characteristics, i.e., patches, along the annotated toolmark edge provides a more robust representation for these data. In the previous section, a similar strategy was evaluated using patches instead of profile segments which did not improve performance. However, for striated toolmarks, the reasoning behind this was to reduce the noise by adding a second dimension that redundantly encodes the same one-dimensional profile. In contrast, for toolmark impressions, both dimensions in the patches contain information about the local characteristic.

This section is divided as follows: firstly, the matching and non-matching local characteristics provided by the FORMS dataset described in Section 3.1 are used to train the original PN-Net described in Section 4.1. The PN-Net allows for a comparison with other similar datasets for local image similarity to estimate the complexity of comparing local impression characteristics. After that, the same data is used to train the Deep Network described in Section 4.1. Subsequently, the whole pipeline is evaluated using the annotated FORMS toolmark images. For modeling the global context, the simple sliding window approach, also used for striated toolmarks, and the more flexible DTW are compared. Similar to the previous section, the metrics published in their papers are used when compared with other methods. Otherwise, the information retrieval metrics MAP and CMS are preferred. Finally, the results achieved and the limitations identified are discussed.

### 5.3.1 Local Image Similarity

In order to evaluate how well the proposed methodology can distinguish local characteristics of impression toolmarks, the matching and non-matching patches provided by the FORMS dataset described in detail in Section 3.1 are utilized. Since this is the first dataset providing images patches of this kind, the evaluation closely follows the one proposed by Balntas [BJTM16]. Therefore, instead of information retrieval metrics, the set-based FPR@95 [MRS08] is calculated. Using the FPR@95 also allows a comparison with results on other local image similarity datasets, like the Photo-Tourism dataset [WB07], and thus an estimation of how *challenging* the task is.

The FORMS dataset provides three different partitionings for this kind of binary “match-

ing” and “non-matching” problem definition. The patches are extracted along the annotated toolmark impression edges with a patch size of either  $32 \times 32$ ,  $64 \times 64$  or  $128 \times 128$  pixels, which defines how much area around the impression is visible. When not specified otherwise, the  $64 \times 64$  patches are used since they provide enough information about the toolmark edge without showing too much area around the edge.

Depending on the partitioning, how the patches are extracted and which are considered matches differs. Firstly, for the FORMS-Locks dataset, patches are extracted along the principle direction of the annotated polyline and are therefore oriented in the same way. Secondly, ten randomly oriented patches are extracted for the FORMS-Locks-RR partitioning for each of these patch locations. For both partitionings, patches on the same location on the toolmark edge are defined as “matching”, which means matching patches can be found in all images showing toolmarks made by the same tool. Assuming, for example, a tool that has been used for two break-ins captured in the FORMS dataset. As such, at least two lock cylinders with toolmarks have been collected for this tool. For both these lock cylinders, there may be multiple toolmarks made with this tool, and each lock cylinder has been captured under 11 different lighting settings. Therefore, a patch on the polyline (i.e., toolmark edge) has at least 10 matching patches at the exact same pixel location in the images captured under different lighting conditions. Additionally, for each toolmark edge in the same image, there is a matching patch at the same distance from the start of the toolmark edge that shows the same local characteristics. Furthermore, each lock cylinder in the crime series has an additional matching patch for each toolmark edge. The third partitioning FORMS-Locks-Lighting-RR explicitly isolates the influence of the different lighting conditions, and thus, “matching” patches are only those from the same pixel location in another toolmark image under a different lighting condition. This partitioning removes the influence of human errors in the annotation process and restricts the matching patches’ variability to lighting differences. “Non-matching” patches are extracted from the same toolmark edge at a different position to avoid trivial solutions due to illumination differences and the structure of the cylinder lock. These patches are expected to show other local characteristics since the local characteristics are assumed to be independent. Yet, this strategy ensures that “hard” triplets are generated for training. These three partitionings of the FORMS dataset allow an evaluation of how well the local characteristics on the edge of the toolmark impression of the same tool can be matched under varying orientations and vastly different lighting conditions. In addition to the FORMS dataset, the baseline PN-Net is also evaluated on the Photo-Tour dataset [WB07] to provide a reference.

The evaluation is conducted with two different CNN architectures. Firstly, the original shallow triplet model PN-Net by Balntas [BJTM16], which is described in detail in Section 4.1.2, is used as a baseline. For the PN-Net input images for the network are  $32 \times 32$  grayscale image patches, and the embedding dimension is fixed to 128. On the Photo-Tour dataset [WB07], which consists of matching and non-matching  $32 \times 32$  images patches extracted from 3D mapped tourist photos with three different subsets (Liberty, Notre-dame, and Yosemite), they achieve a false positive rate at 95% recall (FPR95) of

4-10%; depending on the subsets used for training and evaluation. Secondly, a deep neural network based on the DenseNet [HLvW17] is used to show the impact of increasing the number of parameters and using a more recent network architecture.

In order to train the PN-Net using the FORMS patches, they are downscaled to  $32 \times 32$  (from  $64 \times 64$ ) to use the same network architecture. The best results are achieved on FORMS-Locks-Lighting-RR partitioning with an FPR95 of 31.68%, as shown in Table 5.5. One interpretation for the remaining false positives is that many patches, mainly from locks made out of shiny materials, are indistinguishable due to the limited dynamic range of the images. Even though this result is still not as good as on the Photo-Tourism dataset, it indicates that adapting to the various lighting conditions is the least challenging problem. The results on the FORMS-Locks and FORMS-Locks-RR are worse, with an FPR95 of 78.77% and 83.24%, respectively. However, the difference between these two partitionings is only about 4-5%, which shows that the CNN does not simply learn to distinguish different orientations of the patches, and the most challenging problem in the dataset is the actual matching of patches from different toolmarks.

	PN-Net	DenseNet
FORMS-Locks	78.77%	70.7%
FORMS-Locks-RR	83.24%	83.9%
FORMS-Locks-Lighting-RR	31.68%	19.6%

Table 5.5: FPR95 achieved on the  $64 \times 64$  FORMS patches downscaled to  $32 \times 32$  by the shallow PN-Net compared to the deep DenseNet.

As shown, the results provided by the PN-Net show room for improvement, and the experiments suggest that a deeper network with more parameters might be able to identify matching and non-matching local toolmark patches more accurately. For this, the DenseNet architecture presented in Section 4.1.2 is used in a similar setup. Table 5.5 shows this leads to an improvement of the results for FORMS-Locks from 78.77% to 70.7% FPR95 but shows no improvement for FORMS-Locks-RR. The most significant improvements are achieved on the FORMS-Locks-Lighting-RR partitioning with more than 10%. However, it is crucial to utilize an early stopping strategy as the training FPR95 plateaus after 4 million iterations, and the validation FRP95 increases then again after this to about 34%. A DenseNet with a depth of 40 blocks was used for all these experiments. Similar to the PN-Net setup, the  $64 \times 64$  patches were downscaled to  $32 \times 32$ .

Using the  $64 \times 64$  patches directly without downscaling does not improve the results. For the FORMS-Locks-RR partitioning, this even increases the FRP95 to 87.1%. By decreasing the depth of the network to 20 blocks to fight overfitting, some performance loss can be compensated, yet the overall performance stays the same at 83.8% FRP95.

An additional evaluation is performed with  $96 \times 96$  crops taken from the center of the  $128 \times 128$  FORMS-Locks-RR patches to investigate the influence of the area visible in the patches. In this experiment, the PN-Net achieves an FRP95 of 82.7%, compared to 83.24%, which is slightly better than the results on the  $64 \times 64$  patches and implies that

either the performance could be improved by using a network with more parameters or that just the increased patch size is beneficial.

In summary, it can be noticed that the FPR95 is very high, i.e., above 70%, for the FORMS-Locks and FORMS-Locks-RR partitionings, which suggests that training a network with patches from different impression toolmarks made by the same tool is not working as well as expected either due to the inherent visual difference of the characteristics or due to inaccuracies in the annotation process. Further, even though the FORMS-Locks-RR provides more samples to fight overfitting, it performs noticeably worse, suggesting that it is harder for the model to learn an invariant rotation representation of the local characteristics. Even though the developed annotation tool allows for a precise alignment of the polylines, a pixel-perfect match can not be achieved. In contrast, the results on the FORMS-Locks-Lighting-RR partitioning are promising, especially with the DenseNet model, which achieves an FPR95 of less than 20% even. Additionally, using bigger  $96 \times 96$  crops, which show an increased area around the impression edge, improves the results slightly.

A final evaluation considering these insights is conducted in the following way: firstly, the bigger  $128 \times 128$  patches are used to create a new FORMS-Locks-Lighting partitioning that removes rotational invariance as a constraint since it has shown to hinder the performance and is not required since the angle of the polylines can be used to calculate the rotation for each patch in the matching process, as shown in Section 5.3.2. Furthermore,  $96 \times 96$  center crops are downscaled to  $64 \times 64$ , and a DenseNet with a depth of 20 blocks is used. An FPR95 of 7.36% can be achieved using this setup with continued improvement in the validation FPR95 after more than 12 million training iterations, which supports the notion that the model can learn to distinguish local characteristics of toolmark impressions when trained with the FORMS-Locks-Lighting partitioning. Using the full  $128 \times 128$  patches instead of  $96 \times 96$  center crops leads to a slightly worse performance of 9.5%. Re-introducing random rotations just for the training patches does lead to an unstable training and overfitting with a similar performance of 19.2% FPR95 as achieved on the FORMS-Locks-Lighting-RR shown above.

For all PN-Net experiments, a batch size of 128 was used with an SGD optimizer with weight decay of  $10^{-4}$ , learning rate of 0.1, learning rate decay of  $10^{-6}$  and Nesterov momentum of 0.9. For all experiments with the original PN-Net, the ratio loss  $l_{ratio}$  was used as defined in Section 4.1.2. The DenseNet architectures utilize the simpler margin loss  $l_{margin}$ . The data augmentation was limited to removing a mean pixel value calculated over the entire training set. For the best performing DenseNet models, the rate for the dropout layer was set to 0.5, the reduction to 0.5, 20 blocks with a growth rate of 20. The bottleneck layer was enabled, and no separate fully-connected feature layer at the end was used. The batch size was adjusted to 32 to accommodate the increased memory requirements of the DenseNet. Again, for the optimizer, SGD with weight decay of  $10^{-4}$ , a learning rate of 0.1, a learning rate decay of  $10^{-6}$ , and Nesterov momentum of 0.9 were used. With these options, the DenseNet has even fewer parameters than the shallow PN-Net with 160k vs. 600k, respectively. Even though no separate fully-connected

embedding layer is employed at the last layer of the DenseNet, and thus the embedding dimension is dependent on the growth and reduction factors in the architecture, attention has been given to ensuring the embedding dimension is kept below 128 from the original PN-Net not to provide an unfair advantage.

### 5.3.2 Global Context

The previous section showed that both the shallow PN-Net and the deep DenseNet are unable to reliably distinguish local toolmark characteristics from the FORMS-Locks and FORMS-Locks-RR partitionings. However, by training on the Lighting\* partitionings and thus removing inaccuracies in the annotation process and focusing solely on the different lighting conditions, FPR95 of 32% and less can be achieved. Consequently, the DenseNet trained on the FORMS-Locks-Lighting partitioning, which achieves an FPR95 of 7.36%, is used for matching the full toolmark impressions. This network is used to compute embeddings for each patch on the annotated toolmark edge. As described above, the patch size is scaled down from  $96 \times 96$  to  $64 \times 64$  pixels, and patches are extracted every 8 pixels, which provides for fairly dense sampling with overlap between patches to compensate for some annotation inaccuracies and provide enough data points for subsequent comparison. Since the experiments performed using the FORMS patches showed that directly using the output of the DenseNet without an additional embedding layer benefits performance, the embedding dimension is given by the architecture of the DenseNet alone. This results in an embedding dimension of 80 for each patch for the architecture used. Depending on the toolmark length, this yields, for example, a matrix of  $71 \times 80$ .

As described in Section 4.1.3, two different approaches for modeling the global context are compared, namely a simple sliding window methodology, which tries to find the best match between the embedding matrices of two toolmark edges, and DTW, which allows for flexible matching of the embedding vectors. The  $L_2$  distance is used to calculate the similarity of the embedding vectors, as described in Section 4.1.3.

The evaluation is performed with all images and the corresponding annotated polylines in the test set. Since the results achieved on the extracted patches show that the model is able to adapt to the diverse lighting conditions, for this evaluation, only the images captured with the first (“01”) lighting direction are used. This strategy represents a typical use case for forensic experts, who do not capture multiple images from different lighting conditions but instead use one fixed lighting setting to capture one image and then search a database for a similar toolmark. As this is meant to represent the retrieval process of forensic experts, the evaluation is performed as a closed-set evaluation of a retrieval system. Because it provides an intuitive understanding of the performance achieved, the CMS, as described in Section 5.1, is utilized.

For matching the toolmark impressions, a cumulative match score of about 80%, at a retrieval rate of 20%, can be achieved. Figure 5.9 depicts the cumulative match characteristic for both the approach using a fixed step size of 8 pixels and the DTW



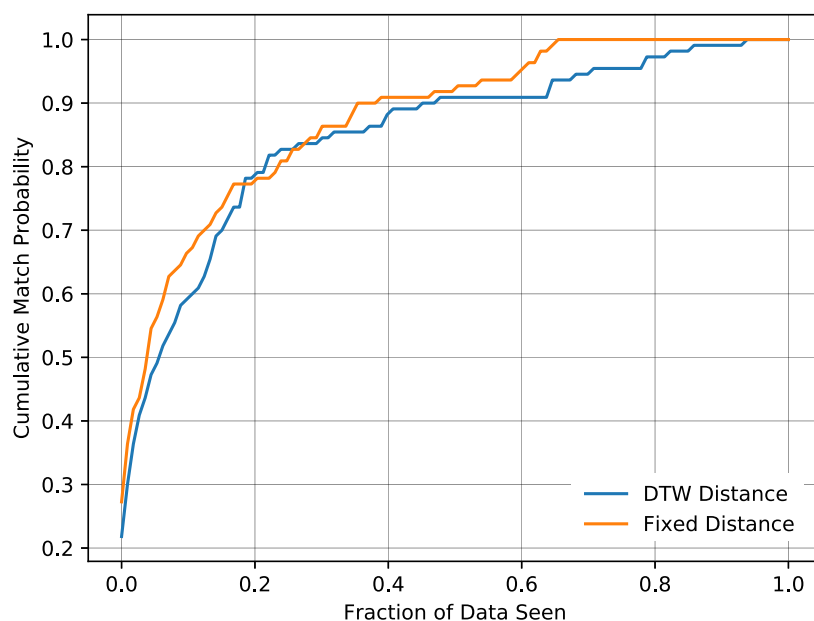


Figure 5.9: Cumulative match characteristic on the test set using either a fixed step size or dynamic time warping. The data is retrieved ranked by similarity.

method. For the annotated FORMS data evaluated, the fixed sliding window approach performs slightly better. These results indicate that the annotations are done precisely, and thus, the increased flexibility of the DTW approach is not beneficial in this instance. Nevertheless, the DTW approach provides comparable results and should therefore be preferred for data that has not been annotated that carefully, as will most likely be the case for data labeled by forensic experts in a time-constrained environment.

Another issue arises from poorly calibrated forensic microscopes. For all the experiments presented, the assumption has been that resolution of the images is precisely the same, which means that all images have the same dots per millimeter (dpmm). However, if forensic microscopes are not calibrated regularly, the actual dpmm of the produced images can change. Therefore, the following experiment with images with a larger and a smaller dpmm value than the training images shows how well the proposed methodology with DTW can handle such scaling errors. A scaling factor of  $227/200$  and its inverse were used to determine how much the results are influence by such changes. As shown in Figure 5.10, the proposed methodology performs best when the dpmm used matches up with the dpmm used to train the model. However, the performance drops for the first few samples retrieved for both scale factors ‘original vs. smaller’ and ‘original vs. bigger.’ For the top-1 accuracy, this results in a difference of about 20%. Nevertheless, after that initial drop, the performance is similar to the ‘original.’ For the experiment with the

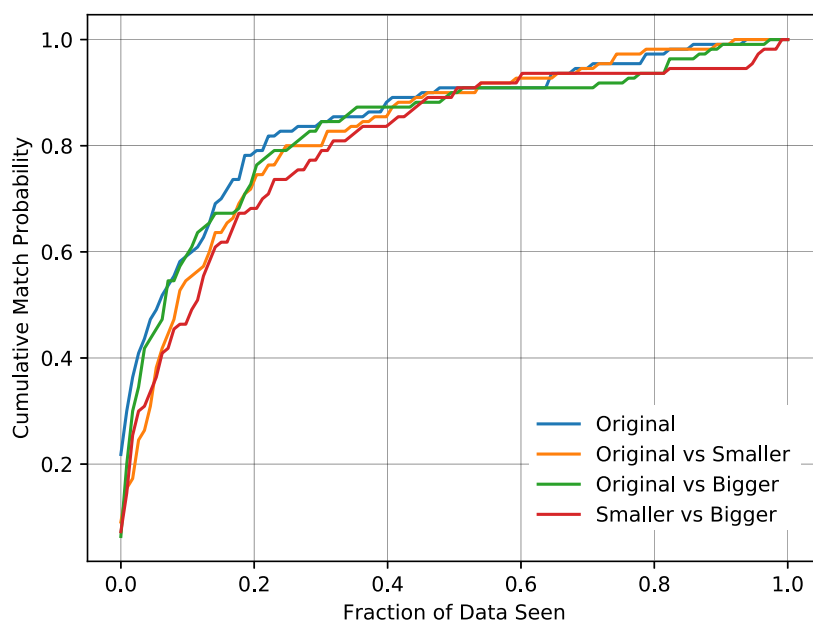


Figure 5.10: Cumulative match characteristic for bigger and smaller dpmm used in the evaluation compared to the training set.

most significant scaling difference, ‘smaller vs. bigger,’ a 10% gap can be observed over the whole CMS curve. This difference suggests that more minor errors in scaling can be compensated by the proposed method, yet the best performance is achieved with regular calibration of the microscopes used by the forensic experts.

### 5.3.3 Discussion

The evaluation conducted in this section shows that the methodology proposed can handle toolmark impressions. The toolmark impressions differ significantly from the striated toolmarks evaluated in the previous section, as they are from real criminal cases, and the local characteristics are two-dimensional. Despite these differences, the methodology utilized for striated toolmarks can successfully be applied to toolmark impressions with only minor modifications in the extraction of local characteristics. Furthermore, all evaluations in this section were conducted as open-set evaluations, which demonstrates that the proposed approach is able to encode local characteristics independently of the sample tools in the training set.

The methodology proposed in this work is the first concerned with such data, according to Baiker et al. [BHK<sup>+</sup>20], and although the performance achieved leaves room for improvement, it is promising and shows that automated retrieval systems can provide valuable support for the work of toolmarks experts. Even though DTW did not immediately lead

to improved performance, modeling the global context in such a way provides a more flexible approach.

## 5.4 Writer Retrieval

In this section, the methodology proposed in Section 4.2 is evaluated using two publicly available datasets. In contrast to the other forensic domains considered in this thesis, writer retrieval has been a focus of the computer vision community, and therefore there are multiple public datasets available. As such, the evaluation is conducted as proposed in previous publications and competitions like the “ICDAR 2013 Competition on Writer Identification” [LGSP13] as an open-set evaluation with *soft-k* and *hard-k* as described in Section 5.1. Additionally, the MAP is used as it also gained some support in recent publications in this field, like [FKD<sup>+</sup>17], and provides a single expressive value to assess the retrieval performance of the testes methodologies.

In contrast to the FORMS dataset evaluated in the previous section, the document samples do not contain annotations for the local characteristics. Consequently, no separate evaluation of the local characteristics and the global context is conducted. Instead, two different approaches for encoding the handwriting style, i.e., the global context, are evaluated, namely the Fischer Vector and the VLAD encoding.

This section is divided as follows: firstly, the ICDAR 2013 and CVL datasets utilized are presented. Secondly, a detailed parameter evaluation is given for the Fisher Vector approach. Subsequently, the improved approach based on the VLAD encoding is presented. For both approaches, strategies for clustering the feature space are investigated. Finally, the results of both proposed methods are discussed and compared to the state of the art.

### 5.4.1 Datasets

Two public datasets are selected to evaluate the proposed writer retrieval methodology, i.e., the “CVL Database” and the “ICDAR 2013” dataset. These datasets provide an open-set evaluation with multiple handwritten pages per writer, and published results by other publications allow a comparison with the state of the art. Furthermore, since handling historic handwritings is not required in the forensic context, datasets with modern handwritings are preferred. To make sure the proposed methodology cannot just “read” the handwriting but rather capture the style of the handwriting, both datasets contain multiple languages. Furthermore, since English, German and Greek are included in these datasets, it can be shown that the proposed methodology can handle different languages and alphabets. Consequently, this allows an assessment of the performance for automatically retrieving handwriting in criminal cases where handwritings in multiple languages might have to be compared. The use of two different datasets allows training on one dataset and evaluation on another, enabling an assessment of how well results can be transferred from one dataset to another. This is additionally motivated by the

reasoning that combining multiple datasets for training might improve the results due to the more significant number of training samples.

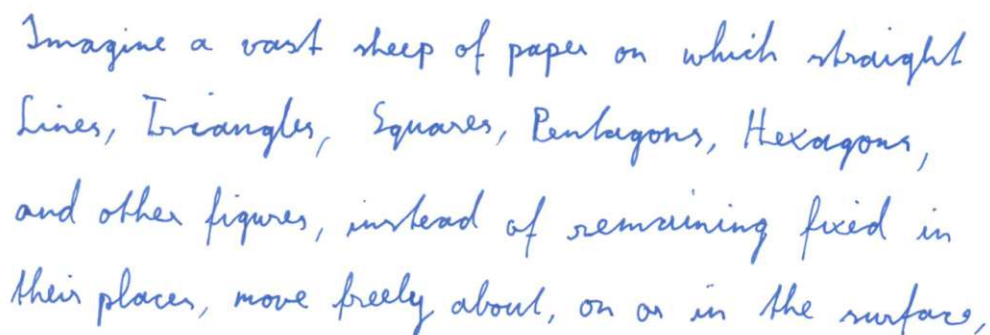
### CVL Dataset

The CVL dataset, also called “CVL Database”, published by Kleber et al. [KFDS13] contains five different handwritten pages by 284 writers provided as 300dpi scanned images. In addition to these three English and two German pages, two additional pages in English are also available for 27 writers. However, as proposed by Kleber et al., only the first five pages of all writers are used to obtain an equally distributed dataset. Even though the pages are supplied as color images, binarization using Otsu’s method [Ots79] is performed to enable a fair comparison with the state of the art, e.g., to [CBA15]. An example page is shown in Figure 5.11.

---

Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said “my universe”: but now my mind has been opened to higher views of things.

---



*Imagine a vast sheet of paper on which straight  
Lines, Triangles, Squares, Pentagons, Hexagons,  
and other figures, instead of remaining fixed in  
their places, move freely about, on or in the surface,*

Figure 5.11: Example page of the CVL Dataset [KFDS13]

### ICDAR 2013

The second dataset used in this section is the dataset of the “ICDAR 2013 Competition on Writer Identification” [LGSP13], short ICDAR 2013 dataset. In contrast to the CVL dataset, it contains multiple alphabets, namely English and Greek, which allows an assessment of how well the notion holds that different handwritings can be distinguished by stroke characteristics instead of character level characteristics. The training set consists of 400 pages written by 100 writers, whereas the evaluation set contains 1,000 pages written by 250 writers. Each author contributed four pages to the dataset, two in English and two in Greek, and each page contains about 2-6 text lines. In Figure 5.12 an exemplary page of the ICDAR 2013 dataset is shown.

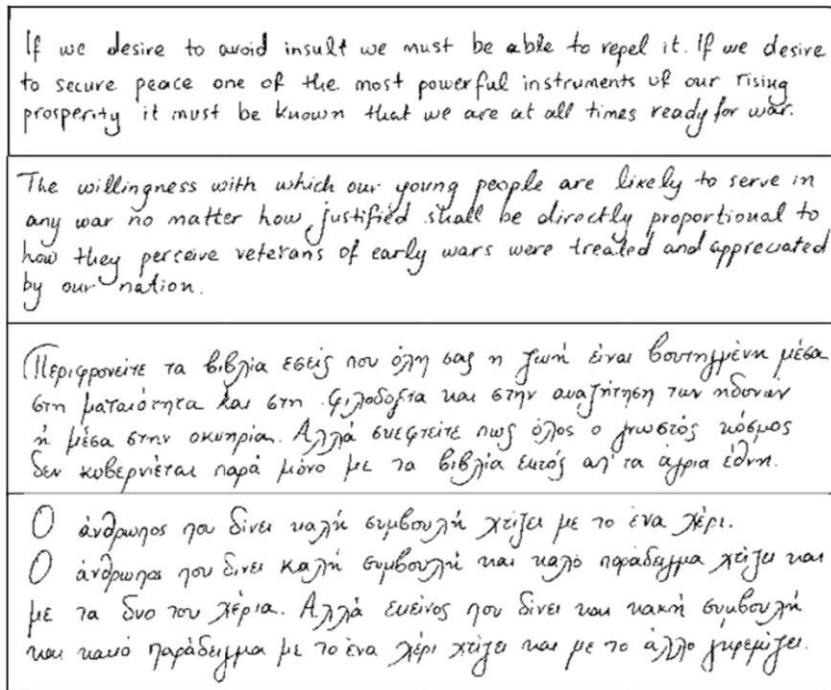


Figure 5.12: Example pages of the ICDAR2013 Dataset [LGSP13]. All four pages have been written by the same writer, two in Greek and two in English.

#### 5.4.2 Fischer Vector

The training of the local characteristics and the global context for these experiments is done with two separate datasets. The triplet CNN is trained using the CVL Dataset, and the Fischer Vector GMMs are constructed by taking the training dataset of the ICDAR13. The reasoning behind this is that the number of training samples is expanded, and it can be shown that the trained CNN is flexible enough to handle completely different data that even has a different alphabet. For these experiments the shallow CNN architecture described in Section 4.2.2 and the SoftPN loss described in Section 4.1.2 are utilized.

To investigate the impact of using a complex model for the global context, the Fisher Vector is compared to the approach of taking the mean of the embedded patches as a feature vector for a document image. Additionally, since Perronin proposed the application of PCA to the SIFT features in [PSM10], also PCA to 32 dimensions is applied to the embedded patches. In these experiments, a dimensionality of 64 and 128 is used for the embedding. Figure 5.13 shows the results on the ICDAR 2013 test dataset. It can be seen that the 64-dimensional features (solid lines) perform better than the 128-dimensional features (dashed lines). Further, whitening of the Fisher Vector boosts the performance, but this gain is more pronounced with less than 20 cluster centers. The dimension of the Fisher Vector after whitening is limited to 1,000; thus, a dimension

Evaluation on the ICDAR 13 dataset with 64 and 128 dimensional features

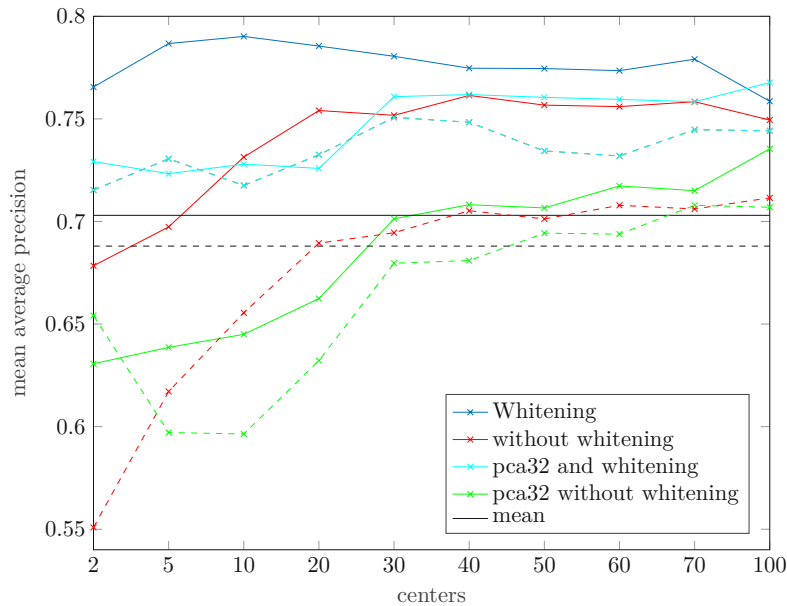


Figure 5.13: Evaluation on the ICDAR13 test dataset with 64 (solid) and 128 (dashed) dimensional embedding. Additionally a PCA to 32 dimensions is also applied to the features. The black lines are the mean average precision when taking the mean of the embedded patches as feature vector for the page.

reduction is performed in the whitening step by using only the largest 1,000 eigenvalues and corresponding eigenvectors. The best performance is achieved using 64-dimensional features and 10 Gaussians. The application of PCA decreases the performance when a low number of Gaussians is used.

If the embedding dimension is set to 128 and only a few Gaussians are used, the Fisher Vector's performance is lower than the performance of the mean. However, with an increasing number of Gaussians, the performance improves. Noticeable is that the MAP of the naive approach of taking the mean is 70.3% and 68.8% with 64 respectively 128 dimensions. These results show that the features successfully encode the writing style even without a complex model.

Further evaluations are carried out with an extended training set for the CNN to investigate the impact of more diverse training data. This training set combines the CVL Dataset and the ICDAR 2013 training set, and the embedding is learned on this dataset, whereas the fitting of the GMMs is done only on the ICDAR 2013 training set. However, this strategy leads to a lower performance than using only the CVL dataset to learn the embedding and the ICDAR 2013 training set for the GMMs. These results might be the consequence of the goal of the neural network that the embedding learned should cover the whole feature space. Thus, writers already form good clusters in the feature space, and the Fisher Vector does not encode additional information. Following

this argument, the datasets for learning the embedding and for clustering the Gaussians should be disjunct. Another explanation is that the triplet network cannot create a homogenous embedding space when using in-homogenous data samples.

The first step of the proposed methodology is to extract the patches from the document image. As mentioned in Chapter 4 two possibilities of extraction are analyzed: firstly, using the SIFT keypoint location, and secondly, using random patches. Experiments show that the extraction of random patches has a lower performance than when using patches extracted at the location of the SIFT keypoints, as seen in Figure 5.14. Interestingly using random patches with whitening leads to a performance decrease which contradicts other findings that whitening improves the performance. One possible explanation for this is that by extracting patches randomly, local characteristics present in all handwritings independent of the writer lead to dimensions of the feature vector that are very similar and thus do not contribute to distinguishing the writing style of a page. Whitening emphasizes these dimensions with low variance, which decreases performance. In contrast, using similar SIFT locations for extracting the local characteristics leads to better features that benefit from whitening. Because of the similarity of the patches, the CNN can learn the similarity, respectively the dissimilarity, better on patches extracted at the SIFT location in contrast to the random patches where the patches are uniformly distributed apart from the threshold of the percentage of stroke pixels introduced. This *normalization* helps to provide better samples for the triplet network to learn more distinctive features that do not depend as much on the location of the extracted local characteristics.

Since taking only the mean embedding of the patches as feature vector already shows good performance with a MAP of 70.3%, and the use of multiple vocabularies is motivated by [JC12], the following is proposed: for each writer, this approach generates GMMs, concatenates the resulting Fisher Vectors and uses this vector as its feature vector for the image. In this way, firstly, multiple vocabularies are created independently. Secondly, since feature representations of the same writer may be scattered in the feature space because the embedding is trained on a different dataset, each vocabulary also encodes the distribution of a particular writer in this embedding. It can be assumed that unknown writers follow similar distributions, and thus some vocabularies can give a decent description of the writing style. When using multiple vocabularies, the dimensionality increases linearly with the number of vocabularies, and a joint dimensionality reduction is applied to remove co-occurrences [JC12]. Table 5.6 shows the results when following this scheme. When using only one Gaussian for each writer, the center of this Gaussian coincides with the mean, but since the Fisher Vectors encode additional information, a performance gain of 8.9% is achieved when also applying whitening. The higher the number of centers per writer, the better the proposed method's performance. However, a maximum of 4 Gaussians for each of the 100 writers in the training set is used. The resulting feature vector for 400 different Gaussians already has a dimensionality of 12,800. Therefore, more Gaussians are not feasible anymore. Similar to the former evaluations, the dimensionality is reduced to 1,000 dimensions in the whitening process since the calculation of the SVD is very time and memory-consuming. When using 3 or 4 Gaussians

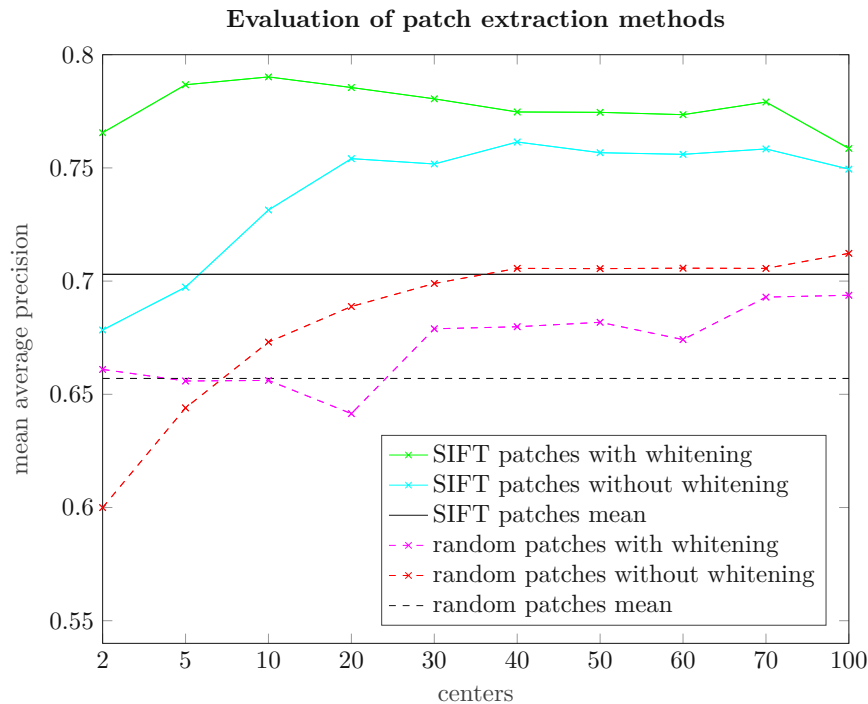


Figure 5.14: Evaluation of both patch extraction methods on the ICDAR 2013 test dataset using 64 dimensions.

for each writer, the proposed methodology achieves the best performance with a MAP of 81.4% after whitening.

centers	without whitening	with whitening
1	72.5	79.2
2	74.7	80.9
3	75.7	<b>81.4</b>
4	76.5	<b>81.4</b>

Table 5.6: Evaluation for increasing number of Gaussians used per writer using the 64 dimensional embedding.

### 5.4.3 VLAD Encoding

Since the method described in the previous section is not able to achieve state-of-the-art results with a MAP of 81.4% on the ICDAR 2013 dataset, in this section, the Fischer Vector encoding is replaced by VLAD as suggested by Christlein et al. [CGFM17]. Even though the proposed method achieves a performance improvement of 14% MAP over Fiel and Sablatnig’s [FS13] utilizing the Fischer Vector and SIFT features on the ICDAR 2013 dataset, the small performance difference compared to averaging the embedding



vectors of 11% suggests that the Fischer Vector is not well suited for encoding the learned embedding vectors. Furthermore, since the results in Section 5.3 suggest that the DenseNet can improve the performance compared to a shallow network, it is utilized for this evaluation. Similarly, as in the previous section, the SoftPN loss is used.

Additionally, in contrast to the evaluation conducted in the previous section, the experiments conducted in this section are carried out exclusively on the “ICDAR 2013 Competition on Writer Identification” [LGSP13] dataset. The results in the previous section show that training with homogenous data may not be beneficial for triplet networks. Therefore, in this section, the ICDAR 2013 training set is used to learn the similarity for the measure of the local characteristics and for creating the vocabularies for encoding the global context. The ICDAR 2013 test set is utilized for evaluation only.

First, the patches on both datasets are extracted, resulting in about 640k and 2.1M patches for the training and evaluation dataset, respectively. For the training of the triplets filter, the patches are filtered using the surrogate classes as described in Section 4.2. This step reduces the number of patches to about 300k. These patches are then used to generate 1.28M triplets for each training epoch. The evaluation with different vocabulary sizes, i.e., the total number of VLAD cluster centers, is conducted to investigate the influence of this parameter. Additionally, a single VLAD vocabulary is compared to using five vocabularies with dynamic sizes derived as described in Section 4.2. As a feature descriptor for each patch, the whitened output of the trained CNN is used. Both the Euclidean and the cosine distance are evaluated since the network is designed to learn a Euclidean metric, and whitened data usually has a good performance when using the cosine distance.

Figure 5.15 shows the MAP on the evaluation dataset, with a varying total number of clusters. Furthermore, multiple VLADs are compared against using a single vocabulary. Additionally, the Euclidean distance and the cosine distance are used. For the training, 100 surrogate classes are used for filtering out patches as described above. This increases the performance compared to taking all patches. The number of surrogate classes was determined empirically by analyzing the results of 50, 100, 500, 1000, and 5000 classes.

The best performance of 86.1% MAP is achieved using 5 VLADs with a total number of 100 cluster centers and the Euclidean distance. Multiple vocabularies outperform a single one in every experiment, particularly when the total number of cluster centers increases. However, this difference is modest for low total numbers of cluster centers. The small individual vocabulary sizes can explain this. For instance, for a total number of 50 centers, the sizes of the five vocabularies are just 25, 12, 6, 3, and 1. Nonetheless, combining whitening to decorrelate the multiple vocabularies is crucial for the performance. Experiments with ten vocabularies, which were done additionally, did not improve the results.

Further, the results show that the Euclidean distance performs better and is more robust to changes in the total number of centers. Since this is not restricted to the usage of multiple VLADs, it suggests that the Euclidean distance is better suited for the proposed

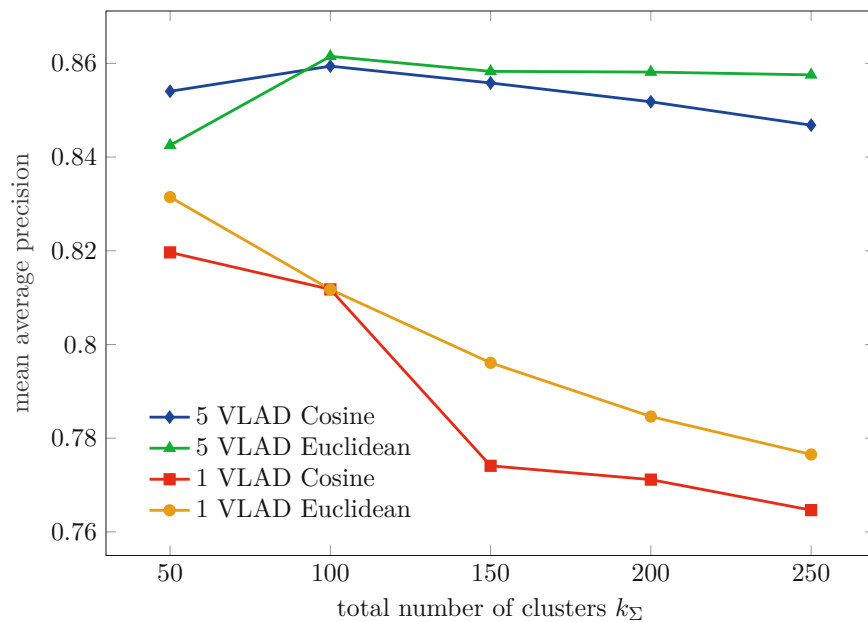


Figure 5.15: Evaluation using the ICDAR13 test dataset with a varying number of clusters and VLADs using cosine and Euclidean distance. In training, 100 surrogate classes were used.

methodology. Additional experiments were conducted with different sizes of the last linear layer of the network, i.e., the feature dimension in the embedding. When lowering the dimension to 64 or 32, the performance drops slightly. Nevertheless, by increasing the last linear layer to 256, the improvements are not significant enough to warrant the doubling of the embedding dimension. These results show that a dimension of 128 is a good trade-off between performance and feature descriptor size.

#### 5.4.4 Comparison & Discussion

This thesis proposes a method for writer identification based on learning an embedding representing the similarity of the handwriting of patches extracted from the document images. The evaluation shows that the learned embedding is expressive enough that even without a Fisher Vector or VLAD encoding, a MAP of 70.3% can be achieved on the ICDAR 2013 dataset by averaging all embedding vectors of a handwritten page.

Nevertheless, by encoding learning embedding vectors the performance can be improved further. Table 5.7 shows the comparison of the method proposed to two other approaches. Christlein et al. [CBA15] have one of the best-performing methods on the ICDAR 2013 dataset. It can be seen that the method proposed based on the Fisher Vector is outperformed by 6.5%. All methods exhibit a performance drop when using the Top 2 criterion. Since all writers have two pages in Greek and two pages in English in the dataset, a document image written in the other language has to be found to achieve

high Top 2 results. However, since the proposed method has a higher performance drop than [CBA15] it can be concluded that the change of alphabet has a more substantial influence here. Nevertheless, this approach builds on Fiel and Sablatnig[FS13] who propose the application of the Fisher Vector to SIFT features for generating the feature vector of an image, and the method proposed achieves a 14.0% better MAP compared to [FS13] which indicates that the learned embedding is well suited to encode the writing style. The approach based on the VLAD encoding performs significantly better with 86.1%, but still slightly worse (2%) than [CBA15],

	MAP	hard		
		Top 1	Top 2	Top 3
Christlein et al. [CBA15]	88.0	99.4	81.0	61.8
Fiel and Sablatnig [FS13]	67.4	94.5	48.0	25.7
proposed Fisher Vector	81.4	95.7	70.8	47.4
proposed VLAD	86.1	98.9	77.9	56.4

Table 5.7: Comparison of the methods proposed to state-of-the Fischer Vector and VLAD approaches on the ICDAR 2013 dataset.

## 5.5 Footwear Impressions

In this section, the methodology for the comparison of footwear impressions presented in Section 4.3 is evaluated. As described in Section 4 modeling the global context is not feasible for some data since it requires detailed annotations. Furthermore, as shown in Section 2 in many applications an end-to-end based approaches can lead to vastly improved performance compared to modeling the problem explicitly. The footwear impression dataset Impress was designed to allow the training of neural networks since it contains a sufficient number of different shoes and multiple images of footwear impressions for each of those shoes. A detailed description of the dataset can be found in Section 3.2.2. The evaluation in this section focuses on the high-quality Inkless Pad impressions, which contain footwear impressions that are clearly separable from the background, and the Wallpaper impressions representing more challenging samples. In contrast to other samples provided by the Impress dataset, these impressions do not need manual preprocessing and allow therefore training and evaluation without human interaction. Hence, these samples are used for training exclusively. The ‘realistic’ impressions, primarily produced using gel lifters, require manual annotation of the area where the footwear impression can be found in the images since the methodology proposed was not designed to automate this step. Therefore, an additional, smaller evaluation dataset was created by selecting diverse samples from the “Realistic Impressions” provided by the Impress dataset. To provide a comparison with the state of the art, the methodology proposed is also evaluated on the FID-300<sup>2</sup> dataset by Kortylewski et al. [KAV15].

<sup>2</sup><https://fid.dmi.unibas.ch/>

This section is divided as follows: first, the implementation details are presented. Secondly, the results on the Inkless Pad and Wallpaper footwear impressions and the realistic impressions from the Impress dataset are shown. The evaluations in these sections include a comparison of two different data augmentation techniques and a preprocessing step for the manually annotated data. Subsequently, a model trained just on the Inkless Pad and Wallpaper impressions is evaluated on the FID-300 dataset and compared to other approaches. Finally, the presented results are summarized.

### 5.5.1 Implementation Detail

In contrast to the previous sections, the network architecture used for the evaluation in this section is a Resnet18 to show that in case enough data is available, the proposed metric learning approach can be applied to other forensic domains in an end-to-end manner without spending too much time on network architectures and parameters. As such, a Resnet18 pre-trained on ImageNet (with an error rate of 30.24%<sup>3</sup>) is used. The images are converted to RGB, if necessary, and normalized with the mean and standard deviation of the ImageNet, as specified by the pre-trained model. The embedding dimension of 64 is ensured with an explicit fully-connected embedding layer that replaces the classification layer in the Resnet. Other parameters are based on suggestions provided by the PyTorch based metric learning Framework<sup>4</sup> by Musgrave et al. [MBL20]. For each batch, 8 labels with 4 samples each are sampled, and triplets are found directly in this batch. The simplified triplet loss function defined in Section 4.1.2 without anchor swap is used. Triplets outside the margin of 0.1 are filtered out using MultiSimilarityMiner [WHH<sup>+</sup>19], described in Section 4.3.2, which is achieved by setting epsilon to the same value as the margin. Two independent Adam optimizers are utilized for the trunk (Resnet18) and the fully-connected embedder to allow different learning rates. A higher learning rate of  $10^{-5}$  is used for the embedder since it is trained from scratch. For the pre-trained Resnet18 trunk, a learning rate of  $10^{-4}$  is used.

The random rotation is performed with the full range of -180 to 180 degrees for both the affine and Euclidean transformation. In the affine case, a scaling factor is randomly picked between 0.9 and 1.0 and for the Euclidean transform from 0.8 to 1.0. The random aspect ratio change used in the affine transformation is done with a factor of 0.75 to 1.33, and for the translation the bounds are set to 0.0 and 0.1.

As the “Realistic Impressions” are segmented using the alpha channel, an additional preprocessing step “UI Preprocessing” is (optionally) added at the beginning, which crops the actual footwear impressions using the alpha channel and places them centered on a new canvas with white background. Without this step, it cannot be guaranteed that the size of the impression is always the same since the resizing function resizes the smaller edge of the image to 256 pixels. Impressions from the FID-300 dataset are pasted into a 586×586 pixel canvas and then treated the same way as the images from the Wallpaper or Inkless Pad impressions.

<sup>3</sup>[https://pytorch.org/hub/pytorch\\_vision\\_resnet/](https://pytorch.org/hub/pytorch_vision_resnet/)

<sup>4</sup><https://github.com/KevinMusgrave/pytorch-metric-learning>

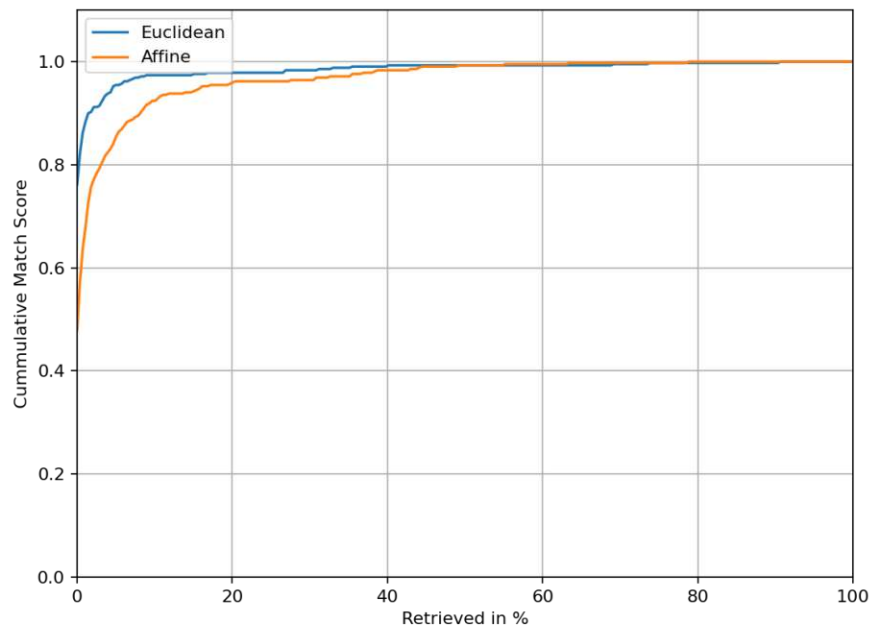


Figure 5.16: CMS wallpaper samples (query) vs Inkless Pad samples (reference database).

The computation effort for computing embedding vectors using an Nvidia GTX TITAN X GPU is about 0.5s per image. This time includes downscaling the image to  $256 \times 256$  and forwarding it through the trunk and the embedder. The comparison of the 64-dimensional embedding vectors can be efficiently performed using the Faiss library<sup>5</sup> and takes about 0.002s after adding a sample to the index.

### 5.5.2 Inkless & Wallpaper Impressions

For these experiments, the Wallpaper and Inkless Pad samples from the second acquisition session in 2019 of the Impress dataset are used as the training set, including 231 pairs of shoes. For each pair, typically 4 Inkless Pad samples (2 left and 2 right) and 6 Wallpaper samples are available. The training is performed with either the Euclidean or affine transformations as data augmentations. The main difference between these strategies is that the affine transformation allows aspect ratio changes, leading to more diverse distortions. Both trained models are used in the subsequent sections without further fine-tuning.

The evaluation is performed using the samples from the first acquisition in 2018 with 69 pairs of shoes. As in the training set, 10 samples per pair are used, i.e., 6 Wallpaper and 4

<sup>5</sup><https://github.com/facebookresearch/faiss>

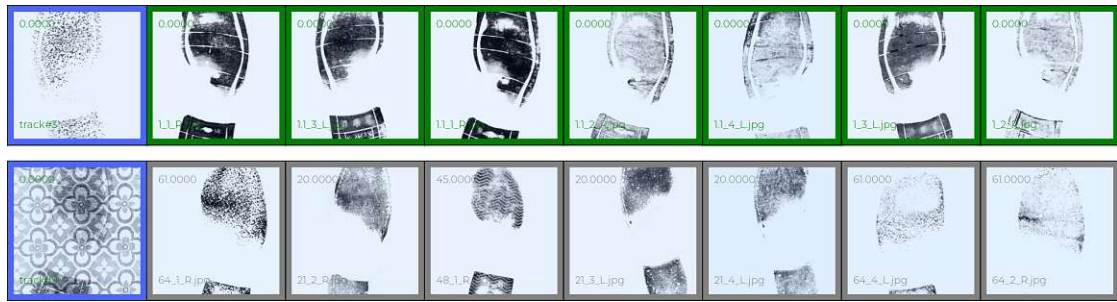


Figure 5.17: Top-ranked Inkless Pad samples retrieved for two Wallpaper queries (blue). Green borders indicate matching and gray non-matching samples.

Inkless Pad samples. These pairs of shoes are not part of the training set for an open-set evaluation setup. Experiments showed that utilizing a leave-one-out strategy with all samples in the whole dataset as queries and in the retrieval document collection results in a non-informative CMS graph as the first retrieved results are almost always the matching impressions of the same kind, i.e., Wallpaper or Inkless Pad. Therefore, for the results shown in Figure 5.16 only the Wallpaper impressions are used as queries, and the Inkless Pad samples are used as the retrievable document collection. As expected, the more restrictive Euclidean augmentations perform significantly better in these experiments. Figure 5.16 shows that for the model trained with affine transformations, the top-1 accuracy is almost 80% compared to about 50% for the Euclidean. Intuitively this makes sense, as the aspect ratio can not change between the impressions since all the impressions were made on a flat surface and captured using a flatbed scanner. In these experiments, the more diverse augmentation enabled by the affine transformations has a negative impact on the performance instead of improving it since the added robustness is not required for this task.

Nevertheless, the results are promising for both models as the chance of retrieving a matching sample out of 1,172 is more than 90% when just 100 samples are retrieved. These results show that the methodology proposed can successfully compare footwear impressions acquired under vastly different conditions, as described in Section 3.2. Figure 5.17 shows an example in the top row in which all top-ranked results (from left to right) are matching samples. However, the bottom row exemplifies that the challenging “special” Wallpaper impression can contain too much background structure, and thus the model fails to find matching samples. Nevertheless, in this case, the first matching sample was found at rank 26, which is still below 2.5% retrieved.

### 5.5.3 Realistic Impressions

For this evaluation, a small dataset was created containing one sample for each of the ten “Realistic Impressions” from the Impress acquisition session 2018, namely newspaper, cardboard, styrofoam, parquet, perforated metal plate, left + right overlapping, 2 Pairs overlapping, smeared, and wet. These impressions are from the same shoe pairs used

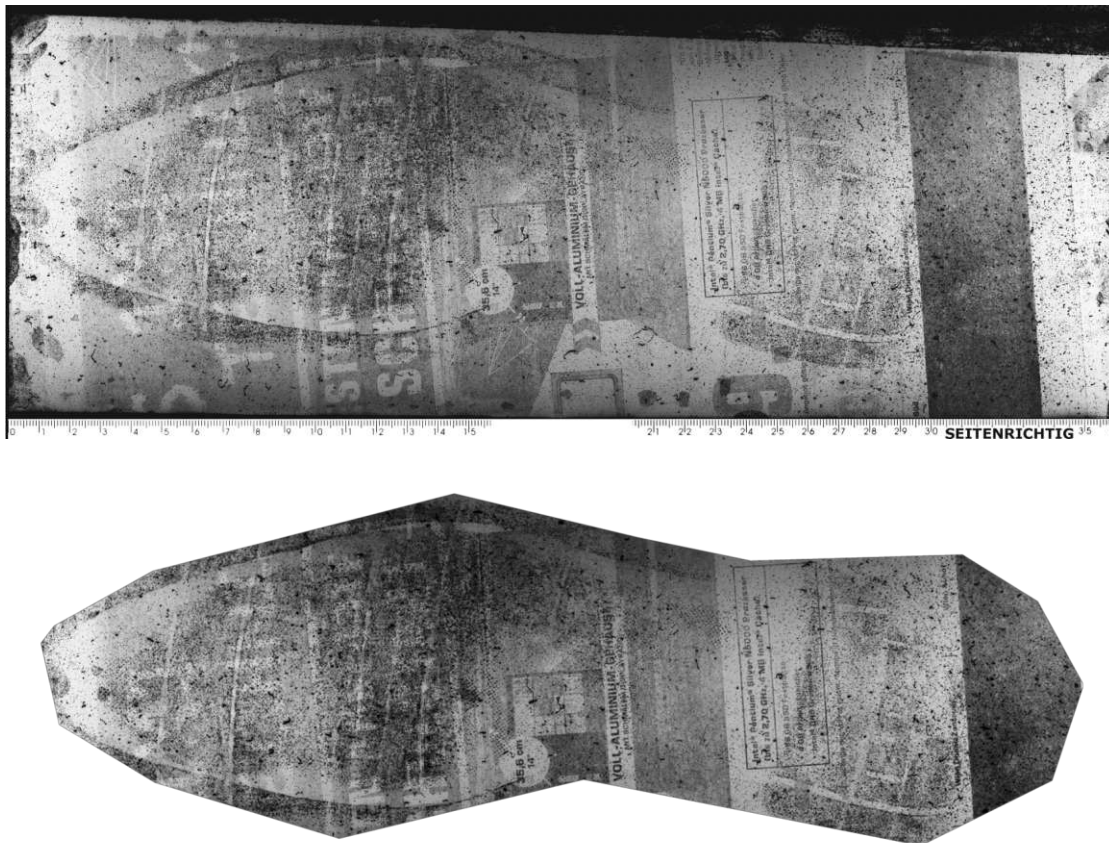


Figure 5.18: Realistic ‘Newspaper’ impressions image with scale (top) which has been manually annotated to just show the area of the impression (bottom).

for evaluation in the previous section and thus do not overlap with the training set. Additionally, two Inkless Pad or wallpaper impressions were added for each of these ten pairs. Impressions from left shoes were manually flipped to look the same as impressions from right shoes.

Even though the resulting dataset with just 30 different samples from ten different pairs of shoes allows only a rough quantitative estimation of the expected performance, it enables a qualitative assessment of issues arising from the samples’ diversity. Figure 5.18 shows an example of the manual preprocessing step. Since these images have been captured with different equipment, the provided scales were utilized to normalize the images’ resolutions.

As realistic impressions are used for this, and manual preprocessing was done using a UI supplied to experts, the evaluation with this dataset represents the results that should be expected when working with actual footwear impressions in the field. As an evaluation metric in this section CMS, as defined in Section 5.1, is used to help communicate the

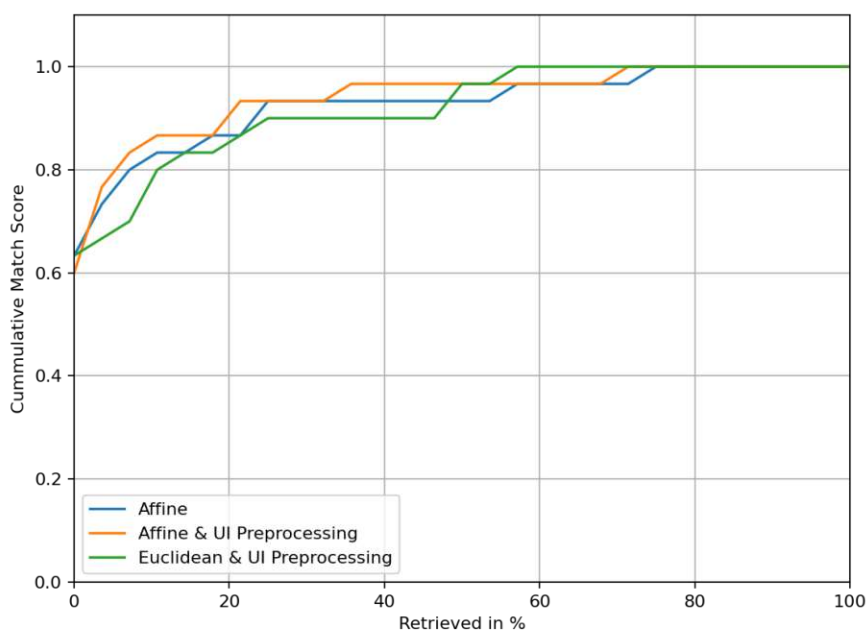


Figure 5.19: CMS for expert data.

results to forensic experts. In order to perform an open-set evaluation, the Impress dataset is split into training and testing sets as proposed in Section 3.2.2 by using the images from the first acquisition session in September 2018 as the testing set and the images from the second acquisition session in 2019 as the training set.

In Figure 5.19 the CMS shows a quantitative assessment of the performance of the proposed methodology on diverse, realistic impressions. Compared to the CMS shown in the previous section, the performance is noticeably worse. The top-1 accuracy drops from about 80% to about 60%, and this performance difference can be observed over the whole graph. However, this performance drop is expected since the samples evaluated contain all the “Realistic Impressions” from the Impress dataset and are more diverse than the samples evaluated in the previous section. Nevertheless, less than 10% of the samples have to be retrieved in order to achieve a CMS of more than 80%. A one-to-one comparison with the results in Figure 5.16 is not possible since the number of relevant samples differs. By employing a different preprocessing strategy, which tries to compensate for the scaling differences between the manually annotated impressions, and the impressions used for training, the results can be improved, which is shown in Figure 5.19 as “UI Preprocessing.” In these experiments the increased robustness of model trained with affine augmentations is shown. The model performs significantly better and achieves an improvement in CMS of up to 15%.



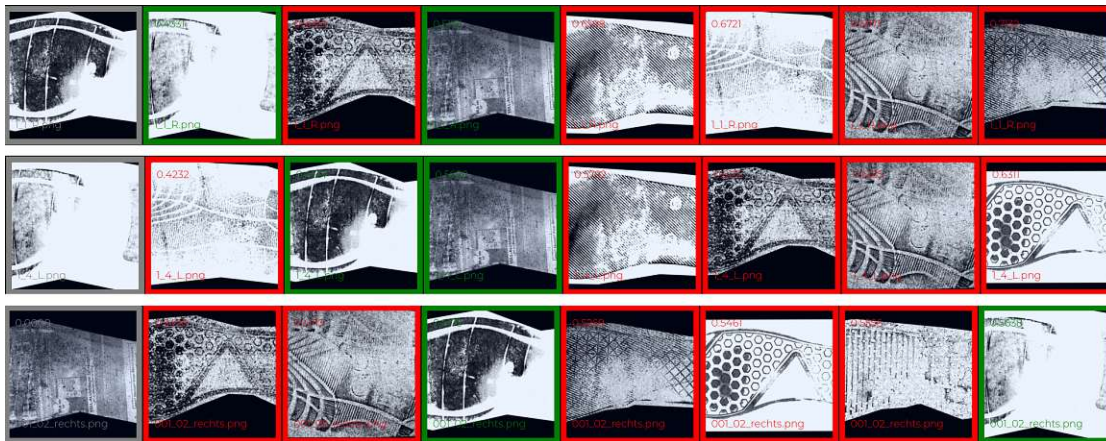


Figure 5.20: Retrieval results for one shoe pair with the query sample framed in gray on the left and the first seven results retrieved from left to right. A green border specifies a matching and a red border non-matching sample.

A qualitative evaluation was performed to investigate which type of samples are retrieved first, which matching samples can not be reliably retrieved, and to identify other issues. This was facilitated by visualizing the first results retrieved for all 30 queries. The results of this qualitative evaluation are shown using selected queries. Figure 5.20 shows the retrieval results for the three samples of the pair of shoes depicted in Figure 5.18. This example shows that, as expected, the matching samples visually most similar to the query sample are retrieved first. In the first row, the query image is an Inkless Pad impression, and the first sample retrieved is the matching Inkless Pad sample. The “Newspaper” impression is retrieved in the third position. The query in the second row shows similar retrieval results. In the last row, the “Newspaper” impression is used as a query, which leads to a significantly worse ranking with the matching samples retrieved on position three and seven. The methodology proposed uses a symmetric distance to calculate distances, i.e., the  $L_2$  distance. Therefore, these results suggest that “Realistic Impressions” from different pairs of shoes form clusters in the embedding space due to other features in the images like background texture. Using k-reciprocal nearest neighbors, as suggested to re-rank the results for person re-identification [ZZCL17] and writer retrieval [RB21], might help to filter out these non-matching samples in a separate post-processing step to mitigate this issue.

In general, it can be noticed that the retrieval works best when the query sample is either a Wallpaper or Inkless Pad impression, which is expected since these kinds of samples are in the training set. Wet impressions and impressions produced by casting also work well as queries since they are visually similar to Inkless Pad impressions. However, the sample size of 30 queries investigated in these experiments is too small to make such assumptions reliably. Nevertheless, these experiments show that even though the training was conducted with just Wallpaper and Inkless Pad impressions, the learned similarity measure is applicable to “Realistic Impression” without fine-tuning or other adjustments

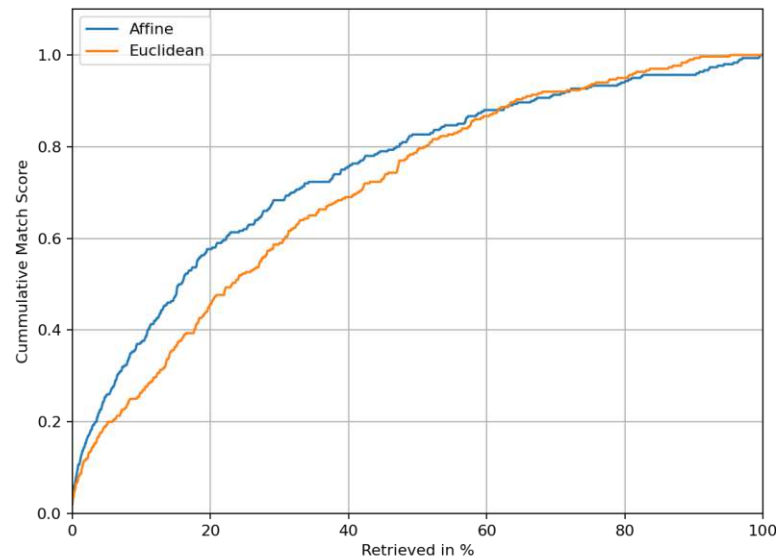


Figure 5.21: CMS for FID-300. Training either random affine or Euclidean transformations.

to the model.

#### 5.5.4 FID-300

This FID-300 dataset [KAV15] provides 1,175 reference images with impressions clearly separable from the background, similar to the Inkless Pad impressions of the Impress dataset. For 300 of these shoe models challenging crime scene impressions, produced by taking images or capturing them using gelatin lifters, were obtained additionally. Since only one or two samples are available for each shoe model, this dataset is only used for evaluation and not for training. In contrast to the Impress dataset, the FID-300 includes partial impressions, which are not explicitly handled by the proposed methodology.

The experiments were performed by using the affine and Euclidean model trained on the Wallpaper and Inkless Pad samples from the Impress dataset, as described above. Except for the preprocessing step, no other changes were made to this model, and no fine-tuning was performed. Figure 5.21 compares the performance of the two models on the FID-300 dataset. Similar to the “Realistic Impressions,” the affine model performs significantly better than the Euclidean. Affine augmentations appear to be beneficial when the model is applied to crime-scene or crime-scene-like images.

In their publication, Kortylewski et al. [KV16] provide an extensive comparison to other approaches using this dataset. Figure 5.22 extends this with the current state of the art by

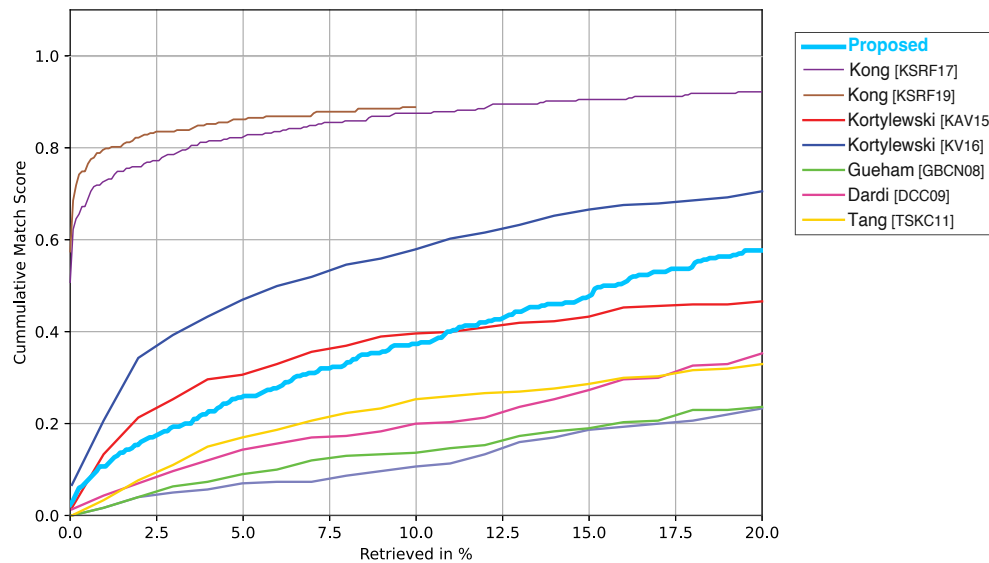


Figure 5.22: CMS for FID-300 compared to state-of-the-art methods.

Kong et al. [KSRF17, KSRF19]. It shows that the methodology proposed is outperformed by both CNN-based cross-correlation approaches by Kong et al. The current approach by Kortylewski et al. [KV16] also performs better, with a performance uplift of about 10% CMS at 20% retrieved. However, the approaches by Kong et al. [KSRF17, KSRF19] are not rotational and translational invariant. Therefore, in addition to  $N \times N$  forward passes through their siamese CNN model, all possible rotations and translations have to be compared to find the best match. This extensive search increases the computational effort considerably, and as such, it is not applicable for the use case considered in this thesis. In contrast, the proposed methodology requires just  $N$  forward passes to compute the embedding vector and an efficient nearest neighbor search in Euclidean space to find matching samples. Similarly, Kortylewski et al. [KV16] need the reference the database to train the active basis models, which are then compared with the query in the evaluation. Figure 5.22 shows that compared to the other methods by Gueham et al. [GBCN08], Dardi et al. [DCC09], and Tang et al. [TSKC11], the methodology proposed performs significantly better.

### 5.5.5 Discussion

The results in this section show that by utilizing the Inkless and Wallpaper samples from the Impress dataset, metric learning models can be trained that not only work with such samples but also with more realistic impressions from the same dataset and with FID-300 samples. In contrast to other state-of-the-art approaches in this field, the proposed

methodology allows an efficient search for similar embedding vectors in Euclidean space using libraries like faiss.

Even though the results are promising, the performance difference to Kong et al. [KSRF19] is significant. Still, the proposed method is faster and more flexible since it is rotationally invariant and invariant (to some degree) to translations, scales, and aspect-ratio changes.

Furthermore, this section showed that the Impress dataset provides enough samples to train a model that can adapt to other datasets. However, since the images were scaled down significantly, the samples' high resolution could not be utilized, and as such individual characteristics, only visible in these high-resolution images, are not considered by the proposed methodology. It also showed that the 'Realistic Impressions' still need manual pre-processing, which is not ideal for including these samples in the training set.

## 5.6 Summary

This chapter evaluated the methodology proposed for comparing forensic images of the domains discussed, namely, toolmarks, footwear impressions, and handwritings. The evaluation was conducted using publicly available datasets and the FORMS and Impress datasets presented in this thesis. First, the problem was defined as an information retrieval task that requires an open-set evaluation with distinct labels for the training and testing set. Subsequently, the metrics used during the evaluation were introduced, including the MAP, which captures the quality over the whole precision/recall curve using one value, and metrics that provide an intuitive understanding of the ranking performance, like hard-k and CMS.

For striated toolmarks, the proposed methodology was evaluated on the NFI Toolmark dataset that contains toolmarks created using screwdrivers at different angles of attack. For this, a detailed evaluation was provided to compare the proposed TripNet with an elastic shape matching baseline and the results published with the NFI Toolmark dataset. Both an open-set and close-set evaluation were performed, and the approaches suggested for extracting the local characteristics were compared.

For impression toolmarks, the evaluation was conducted on the FORMS datasets on both the matching and non-matching local patches and the annotated toolmark images. The evaluation compared the shallow PN-Net with the deep DenseNet on the three partitionings provided by the FORMS dataset. The results showed that removing the inaccuracies in the annotation process and focusing solely on the different lighting conditions was beneficial and allowed the deep DenseNet to utilize its architectural advantage.

In the case of handwritings, the open-set evaluation was performed using the ICDAR 2013 and CVL datasets. Both the extraction of local characteristics and the modeling of the global context using VLAD and Fischer Vector encoding were evaluated in detail.

Finally, an evaluation of the end-to-end base methodology for footwear impression retrieval was provided. An open-set evaluation was performed using Wallpaper and Inkless Pad

impressions from the Impress dataset. Furthermore, manually preprocessed realistic impressions were utilized to demonstrate the performance expected when working with actual footwear impressions in the field. Additionally, the FID-300 dataset was used to provide a comparison with the state of the art.



## Conclusion

This thesis presents a methodology for learning a similarity measure for retrieving forensic images. I analyze the challenges presented by these images on three selected forensic image domains, toolmarks, footwear impressions, and handwritings, for which law enforcement agencies, i.e., the Austrian Police, desire an automatic retrieval. To allow fast retrieval, I propose using a metric-learning-based methodology that utilizes shallow or deep CNNs with a triplet loss function to learn an embedding that allows to implement a computationally efficient distance metric, and thus an efficient search for similar images. Image features are described by local and global characteristics. Since matching local characteristics is crucial for forensic images, the methodology focuses on finding local image similarities. As the global context is specific to each forensic domain, I propose different approaches for extracting local characteristics, like randomly selecting patches, using prior information, and an end-to-end training strategy. For aggregating the extracted local characteristics, i.e., modeling the global context, I propose different methods that describe their positional relationship using a rigid (sliding window) or flexible (DTW) structure, or encode their distribution by clustering the embedding space. Furthermore, I propose an end-to-end CNN-based encoding for footwear impressions.

In order to train and evaluate the proposed metric-learning-based methodology, I created two datasets with impression toolmarks and footwear impressions based on workflows of the Austrian Police to allow a realistic assessment of the expected performance. I designed these datasets explicitly for the training and evaluation of learning-based approaches. Therefore, they contain separate training and testing sets for an open-set evaluation and multiple samples per tool or shoe to capture the variations among impressions created by the same object. The diversity of the Impress footwear impression dataset allows utilizing it in various evaluation schemes. Furthermore, the detailed description of the dataset's creation shows how 4,000 forensic images can be captured efficiently using a well-designed acquisition process. I employed similar techniques for a time-efficient acquisition of 3,046 images of toolmark impressions from real criminal cases that make

the FORMS dataset the first of its kind. This dataset is especially useful for investigating local image similarity in forensic images since it contains manual annotations of matching local areas. Both datasets are publicly available to encourage further research in these fields.

The evaluation shows that the methodology proposed allows fast retrieval of forensic images even though it cannot always achieve state-of-the-art results. For striated toolmarks, the proposed methodology can adapt to variances in the angle of attack between  $15^\circ$  and  $60^\circ$ , achieving a MAP of 78% outperforming the elastic shape matching baseline by 31%. Furthermore, by uncoupling the local characteristics from the global context, the proposed methodology can be extended as a similarity measure for unseen tools (open-set evaluation). Using this uncoupling of local characteristics, I show that the proposed methodology can also be used for impression toolmarks. For comparing local impression characteristics from different lighting directions, the embedding learned using a DenseNet network architecture achieves an FPR95 of 19.6%, and for the retrieval of toolmarks impressions in the FORMS dataset a cumulative match score of about 80% at a retrieval rate of 20% is achieved. Likewise, the proposed embedding learned using patches extracted from handwritten pages is expressive enough to achieve a MAP of 70.3% by just averaging the embedding vectors. Encoding these vectors using Fischer Vector or VLAD improves the MAP to 81.4% and 86.1%, respectively, which is slightly worse than the methodology proposed by Christlein et al. [CBA15] using the VLAD encoding (88.0%) and significantly better than the Fischer-Vector-based approach by Fiel and Sablatnig (67.4%) [FS13]. In contrast to toolmarks and handwritings, for footwear impressions, I propose a methodology that utilizes an end-to-end trained ResNet architecture with a triplet loss function. This approach works well for Inkless Pad reference impressions and the more diverse Wallpaper impressions achieving a top-1 accuracy of almost 80%. For a small dataset with manually processed realistic footwear impressions, similar to impressions from real criminal cases, less than 10% of the samples have to be retrieved to achieve a CMS of more than 80%. However, on the publicly available footwear impression dataset FID-300 the proposed approach is outperformed by the computationally extensive approaches by Kong et al. [KSRF17, KSRF19] and Kortylewski et al. [KV16]. Nevertheless, the performance is better than other methods by Gueham et al. [GBCN08], Dardi et al. [DCC09], and Tang et al. [TSKC11], and I trained the model only on Inkless Pad and Wallpaper impressions. Furthermore, the method proposed is fast and flexible since it is rotationally invariant and invariant (to some degree) to translations, scales, and aspect-ratio changes. In contrast, Kong et al. [KSRF17, KSRF19] and Kortylewski et al. [KV16] require a time-consuming template-matching-like dense search for the best matching rotation and translation.

## Research Questions

In Chapter 1, I define the first of three research questions as “*Can deep learning allow a shared methodology for finding image similarities in different forensic domains?*”. Even though I have to answer this question with “no”, it was clear from the beginning of my



---

research that it was meant to be more aspirational than an outcome of this research. As I showed in Chapter 4, even though the underlying problem of finding local characteristics that in a global context describe forensic images is very similar, as soon as constraints need to be defined to model the problem more accurately, these constraints need to fit the specific forensic domain. The constraints I propose for striated and impression toolmarks, handwritings, and to a lesser degree, footwear impression have two closely connected goals.

Firstly, I employ constraints to utilize the available training data more efficiently. The two provided datasets show that acquiring training data, particularly in forensic domains, is time-consuming. Likewise, the number of real forensic image samples from criminal cases is limited by the number of such cases. Even though I would prefer a lot of serial offenders committing burglaries in Vienna from a data perspective, this is gladly not the case. Secondly, I try to incorporate my knowledge about the problem to prevent the neural network from learning something wrong and nudge the network to learn something right. For instance, for writer retrieval, the similarity of two handwritten documents must be defined by the similarity of the writers' handwriting style and not by the written text or the background. In this example, chopping up a page into unreadable patches is a crude but effective way to achieve this goal. Nevertheless, in case a writer retrieval dataset with hundreds of handwritten pages of the same text per writer is available, a naive metric learning approach with complete pages might also work, which exemplifies that the underlying reason for these constraints is also an efficient utilization of the available training data.

In contrast to these constraints that limit the sample space to make finding a solution easier, augmentations, like random rotations, are utilized. This expands the sample space to find a more general solution that fits the unconstrained sample space of the real world better. Therefore, constraints and augmentations are two approaches for circumventing the problem of a training sample space that does not sufficiently capture the sample space of the real world.

Besides these differences, the network architectures used throughout this thesis are consistently either a shallow CNN or a ResNet (or ResNet-like) deep CNN and contain at most one fully connected layer. Likewise, all loss functions I employ are some variant of the triplet loss. Metric learning might still allow a shared methodology for finding image similarities in different forensic domains using an end-to-end approach given enough training data. Nevertheless, for the data I consider in this work, the methodology proposed has significant differences between toolmarks, handwritings, and footwear impressions.

However, having this lofty goal of a universal methodology always in the back of my mind helped me to consistently think about the answer to the second research question, which reads "*What are the shared concepts of the examined forensic image modalities?*". The most apparent similarity between all the forensic domains I consider in this thesis is the emphasis on local characteristics. In contrast to tasks like object classification, which requires a robust global analysis if the image shows a cat or a house, for forensic images local characteristics are crucial to identify the individual characteristics of a tool,

shoe, or writer. Of course, other characteristics can also be helpful. For instance, the methodology for footwear impressions completely ignores such fine-grained individual characteristics and focuses on model characteristics, similar to finding out which of 147 different dog breeds is present in an image [DDS<sup>+</sup>09]. However, for footwear impressions, this only makes sense since the proposed methodology already filters the retrieved images considerably, and as with all other forensic domains, the final decision is left to the forensic expert.

Another common issue I identified with forensic images is finding and separating the crucial parts that must be analyzed, e.g., the toolmark, footwear impression, which is challenging due to the small scale of individual characteristics. In particular, footwear impressions captured with gelatin lifters are often difficult to separate from the background since the patterns are only made up of areas of dust and areas of missing dust, and any dirt on the floor or the shoe sole is also lifted. Even though this is not as problematic for the handwritten documents I have considered, forensic evidence is generally not created on purpose but as a side product of a criminal act, and therefore, its identification is challenging. Since the primary source of forensic images is law enforcement agencies with restrictive laws regarding sharing forensic evidence, particularly evidence that can be used to identify a person, publishing such images is often not possible, which explains the lack of such datasets. For instance, even though the Austrian Police has an extensive collection of handwritten documents from criminal cases and handwritten documents written by inmates, a publication of these documents is not possible. Similarly, for the Impress dataset, I am legally not allowed to publish footwear impressions from real criminal cases.

Furthermore, forensic experts have to defend their findings in court, and it is ultimately their decision alone if the forensic evidence in question matches or not. Therefore, the goal is to help forensic experts identify similar forensic images more efficiently and not provide a binary matching or non-matching decision. Thus, an algorithm that presents visually similar results is more helpful and more readily accepted by forensic experts than an algorithm that measurably provides accurate results but can not be understood by the experts. In particular, these users often see machine-learning-based methods as black boxes.

For the third research question - *“How can the characteristics in the images be represented to enable an efficient search and comparison in databases?”* - I propose different approaches in Chapter 4. Generally, the proposed approaches for toolmarks, handwritings, and footwear impressions, are based on metric learning. Therefore, on a local scale, a fast comparison is guaranteed, and for footwear impressions, this extends to comparing the whole image since I utilize an end-to-end trained model for these images. After the initial time-consuming training of the model, during inference, each impression image is only forwarded once through the network, and the Euclidean distance can be utilized to search for similar impressions efficiently. This search is sped up further by using indexing libraries like faiss<sup>1</sup>. Additionally, since I trained the model using augmentations, it is

<sup>1</sup><https://github.com/facebookresearch/faiss>

---

rotationally invariant and invariant to other minor translation, scaling, and aspect ratio changes. This further speeds up the comparison since an embedding vector represent an impression and all other augmented versions of this impression, and therefore, no deep search using different configurations, e.g., orientations, is needed. For toolmarks, this works analogously but only on a local scale. The global context is compared similarly to a template-matching approach by comparing all possible configurations. Nevertheless, since the toolmarks are annotated and thus represented by a line or poly-line, this constrained search space can be probed efficiently. Finally, for writer retrieval, the distribution of the local characteristics is encoded in one vector to allow a fast comparison of handwritten documents. Therefore, my arguably unsatisfactory answer to this question is that it can be done efficiently, but the “how” depends on the domain. Selecting a core metric learning methodology that allows a fast search already helps a lot since even if a dense comparison of local characteristics is required, these comparisons are efficient. For the global context, techniques like Fischer Vector and VLAD can be utilized, and in case enough training data is available, an end-to-end approach can be used.

## Future Work

For future work, various improvements can be investigated to increase the performance in the respective forensic domains. For toolmark impressions, a fully automated approach that automatically locates the toolmark impression edges reliably in the images would simplify the workflow of forensic experts significantly. Furthermore, even though DTW did not immediately improve performance, it can be evaluated if the added flexibility improves the performance in the field where the edges of the impressions might not be annotated precisely. Generally, a more expressive global context model might better represent higher-level information. For striated toolmarks, a new dataset with more samples would help train such a model and provide a means for a detailed evaluation. For writer retrieval, firstly, the patches extracted can be examined based on the contained information to improve the expressiveness of the extracted patches. Furthermore, variation in stroke width and font height can be considered, improving performance when changing the dataset. For footwear impressions, as a first step, all the “Realistic Impressions” can be manually or automatically preprocessed to make them available for training. Training with these images would probably improve the performance of samples from real criminal cases used in the FID-300 dataset. As a second step, utilizing learned linear mappings representing different kinds of impressions as used by Kong et al. [KSRF19] might also improve the results on such data. Additionally, a way to handle partial impressions has to be investigated, for instance, by dividing the impressions into multiple overlapping areas. Registering the images similarly to the FORMS dataset would provide a flexible way to divide the image into matching areas. This would also allow the training and evaluation of local characteristics. Furthermore, it would allow training with aligned impressions, improving the results on such preprocessed data by sacrificing flexibility. In general, the Impress dataset can be utilized better, for example, by using the available high-resolution images to compare individual characteristics of the footwear impressions.

## 6. CONCLUSION

---

In addition to these domain-specific improvements, an adaptation of the proposed approach to allow a definition of the constraints in the network architecture is an extensive topic for future work. An example of this is the incorporation of the VLAD encoding as a network layer, as shown by Arandjelovic et al. [AGT<sup>+</sup>16]. Furthermore, this could, for instance, be done using attention mechanisms or by replacing the CNN architecture with a transformer-based model like [DBK<sup>+</sup>20]. Similarly, machine-learning-based methods that focus on explainability can be an extension of this work as well as a user study analyzing the acceptance of the proposed methodology and user experience for such a retrieval system.





# Bibliography

- [AGT<sup>+</sup>16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307. IEEE, 2016.
- [AH08] Gharsa AlGarni and Madina Hamiane. A novel technique for automatic shoeprint image retrieval. *Forensic Science International*, 181(1–3):10–14, 2008.
- [ALV11] Vlad Atanasiu, Laurence Likforman-Sulem, and Nicole Vincent. Writer Retrieval - Exploration of a Novel Biometric Scenario Using Perceptual Features Derived from Script Orientation. In *2011 International Conference on Document Analysis and Recognition (ICDAR)*, pages 628–632. IEEE, 2011.
- [ANYE18] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. MesoNet: A Compact Facial Video Forgery Detection Network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.
- [BBFR13] David Baldwin, John Birkett, Owen Facey, and Gilleon Rabey. *The Forensic Examination and Interpretation of Tool Marks*. John Wiley & Sons, 2013.
- [BC93] Pierre Baldi and Yves Chauvin. Neural Networks for Fingerprint Recognition. *Neural Computation*, 5(3):402–418, 1993.
- [BC94] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, volume 10, pages 359–370. AAAI, 1994.
- [BGL<sup>+</sup>94] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature Verification using a "Siamese" Time Delay Neural Network. In *Advances in Neural Information Processing Systems*, volume 6, pages 737–744. Morgan-Kaufmann, 1994.

- [BHK<sup>+</sup>20] Martin Baiker-Sørensen, Koen Herlaar, Isaac Keereweer, Petra Pauw-Vugts, and Richard Visser. Interpol review of shoe and tool marks 2016-2019. *Forensic Science International: Synergy*, 2:521–539, 2020.
- [BJJK10] Benjamin Bachrach, Anurag Jain, Sung Jung, and Robert D. Koons. A Statistical Validation of the Individuality and Repeatability of Striated Tool Marks: Screwdrivers and Tongue and Groove Pliers. *Journal of Forensic Sciences*, 55(2):348–357, 2010.
- [BJTM16] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors. *CoRR*, abs/1601.05030, 2016.
- [BKP<sup>+</sup>14] Martin Baiker, Isaac Keereweer, René Pieterman, Erwin Vermeij, Jaap van der Weerd, and Peter Zoon. Quantitative comparison of striated toolmarks. *Forensic Science International*, 242:186–199, 2014.
- [BLVM17] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5173–5182. IEEE, 2017.
- [BMI22] Bundeskriminalamt BMI. Polizeiliche Kriminalstatistik (PKS). <https://www.bundeskriminalamt.at/501>, May 2022.
- [BPG<sup>+</sup>16] Martin Baiker, Nicholas D.K. Petraco, Carol Gambino, René Pieterman, Peter Shenkin, and Peter Zoon. Virtual and simulated striated toolmarks for forensic applications. *Forensic Science International*, 261:43–52, 2016.
- [BPZ15] Martin Baiker, René Pieterman, and Peter Zoon. Toolmark variability and quality depending on the fundamental parameters: Angle of attack, toolmark depth and substrate material. *Forensic Science International*, 251:40–49, 2015.
- [BRPM16] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press, 2016.
- [BS07] Marius Bulacu and Lambert Schomaker. Text-Independent Writer Identification and Verification Using Textural and Allographic Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):701–717, 2007.
- [CBA15] Vincent Christlein, David Bernecker, and Elli Angelopoulou. Writer Identification Using VLAD Encoded Contour-Zernike Moments. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 906–910. IEEE, 2015.



- [CGFM17] Vincent Christlein, Martin Gropp, Stefan Fiel, and Andreas Maier. Unsupervised Feature Learning for Writer Identification and Writer Retrieval. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–997. IEEE, 2017.
- [CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546. IEEE, 2005.
- [CHX<sup>+</sup>19] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1861–1870. IEEE, 2019.
- [CM18] Vincent Christlein and Andreas Maier. Encoding CNN activations for writer recognition. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 169–174. IEEE, 2018.
- [CMK<sup>+</sup>10] L. Scott Chumbley, Max D. Morris, M. James Kreiser, Charles Fisher, Jeremy Craft, Lawrence J. Genalo, Stephen Davis, David Faden, and Julie Kidd. Validation of tool mark comparisons obtained using a quantitative, comparative, statistical algorithm. *Journal of Forensic Sciences*, 55(4):953–961, 2010.
- [CNS<sup>+</sup>19] Vincent Christlein, Anguelos Nicolaou, Mathias Seuret, Dominique Stutzmann, and Andreas Maier. ICDAR 2019 competition on image retrieval for historical handwritten documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1505–1509. IEEE, 2019.
- [CS14] Jun Chu and Sargur Srihari. Writer Identification Using a Deep Neural Network. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP)*, pages 31:1–31:7. ACM, 2014.
- [CSS<sup>+</sup>19] Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Král, and Andreas Maier. Deep generalized max pooling. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1090–1096. IEEE, 2019.
- [CTSV13] Wei Chu, Robert M. Thompson, John Song, and Theodore V. Vorburger. Automatic identification of bullet signatures based on consecutive matching striae (CMS) criteria. *Forensic Science International*, 231(1):137–141, 2013.
- [DBK<sup>+</sup>20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

- [DCC09] Francesca Dardi, Federico Cervelli, and Sergio Carrato. A Texture Based Shoe Retrieval System for Shoe Marks of Real Crime Scenes. In *International Conference on Image Analysis and Processing (ICIAP)*, pages 384–393. Springer, 2009.
- [DDS<sup>+</sup>09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.
- [dFR05] Philip de Chazal, John Flynn, and Richard B. Reilly. Automated Processing of Shoeprint Images Based on the Fourier Transform for Use in Forensic Science. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):341–350, 2005.
- [DGXZ19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699. IEEE, 2019.
- [EZG<sup>+</sup>14] Laura Ekstrand, Song Zhang, Taylor Grieve, L. Scott Chumbley, and M. James Kreiser. Virtual Tool Mark Generation for Efficient Striation Analysis. *Journal of Forensic Sciences*, 59(4):950–959, 2014.
- [FDB14] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor Matching with Convolutional Neural Networks: A Comparison to SIFT. *CoRR*, abs/1405.5769, 2014.
- [FKD<sup>+</sup>17] Stefan Fiel, Florian Kleber, Markus Diem, Vincent Christlein, Georgios Louloudis, Nikos Stamatopoulos, and Basilis Gatos. ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI). In *2017 14th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1377–1382. IEEE, 2017.
- [FS12] Stefan Fiel and Robert Sablatnig. Writer Retrieval and Writer Identification Using Local Features. In *2012 10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 145–149. IEEE, 2012.
- [FS13] Stefan Fiel and Robert Sablatnig. Writer Identification and Writer Retrieval Using the Fisher Vector on Visual Vocabularies. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 545–549. IEEE, 2013.
- [FS15] Stefan Fiel and Robert Sablatnig. Writer Identification and Retrieval using a Convolutional Neural Network. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 26–37. Springer, 2015.

- [FVNT15] Nir Finkelstein, Nikolai Volkov, Yehuda Novoselsky, and Tsadok Tsach. A Physical Match of a Metallic Chip Found on a Bolt Cutters' Blade. *Journal of Forensic Sciences*, 60(3):787–789, 2015.
- [GBCN08] Mourad Gueham, Ahmed Bouridane, Danny Crookes, and Omar Nibouche. Automatic Recognition of Shoeprints Using Fourier-Mellin Transform. In *2008 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 487–491. IEEE, 2008.
- [GBJ13] George Gerules, Sanjiv K. Bhatia, and Daniel E. Jackson. A survey of image processing techniques and statistics for ballistic specimens in forensic science. *Science & Justice: Journal of the Forensic Science Society*, 53(2):236–250, 2013.
- [GD18] David Güera and Edward J. Delp. Deepfake Video Detection Using Recurrent Neural Networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.
- [GHRS04] Jacob Goldberger, Geoffrey E. Hinton, Sam Roweis, and Russ R. Salakhutdinov. Neighbourhood Components Analysis. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.
- [HA15] Elad Hoffer and Nir Ailon. Deep Metric Learning Using Triplet Network. In *International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*, pages 84–92. Springer, 2015.
- [HCL06] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.
- [HKB<sup>+</sup>14] Ville Heikkinen, Ivan Kassamakov, Claude Barbeau, Sami Lehto, Tapani Reinikainen, and Edward Hægström. Identifying Diagonal Cutter Marks on Thin Wires Using 3D Imaging. *Journal of Forensic Sciences*, 59(1):112–116, 2014.
- [HLvW17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2017.
- [HSK<sup>+</sup>12] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *In Proceedings of the IEEE Conference*

on *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

- [IS15] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.
- [JC12] Hervé Jégou and Ondrej Chum. Negative Evidences and Co-occurrences in Image Retrieval: The Benefit of PCA and Whitening. In *European Conference on Computer Vision (ECCV)*, pages 774–787. Springer, 2012.
- [JD13] Rajiv Jain and David Doermann. Writer Identification Using an Alphabet of Contour Gradient Descriptors. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 550–554. IEEE, 2013.
- [JFR07] Anil K. Jain, Patrick Flynn, and Arun A. Ross. *Handbook of Biometrics*. Springer Science & Business Media, 2007.
- [JPD<sup>+</sup>12] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.
- [KAV15] Adam Kortylewski, Thomas Albrecht, and Thomas Vetter. Unsupervised Footwear Impression Analysis and Retrieval from Crime Scene Data. In *Asian Conference on Computer Vision (ACCV 2014) Workshops*, pages 644–658. Springer, 2015.
- [KB19] Mahmut Kaya and Hasan Sakir Bilge. Deep Metric Learning: A Survey. *Symmetry*, 11(9):1066, 2019.
- [KCS14] Rajesh Kumar, Bhabatosh Chanda, and J.D. Sharma. A novel sparse model based forensic writer identification. *Pattern Recognition Letters*, 35:105–112, 2014.
- [KFDS13] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 560–564. IEEE, 2013.
- [KFS18] Manuel Keglevic, Stefan Fiel, and Robert Sablatnig. Learning Features for Writer Retrieval and Identification using Triplet CNNs. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 211–216. IEEE, 2018.

- [KS16] Manuel Keglevic and Robert Sablatnig. Learning a Similarity Measure for Striated Toolmarks using Convolutional Neural Networks. In *7th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pages 1–6. IET, 2016.
- [KS17a] Manuel Keglevic and Robert Sablatnig. FORMS-Locks: A Dataset for the Evaluation of Similarity Measures for Forensic Toolmark Images. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1890–1897. IEEE, 2017.
- [KS17b] Manuel Keglevic and Robert Sablatnig. FORMS – Forensic Marks Search. In *Proceedings of the OAGM & ARW Joint Workshop 2017*, pages 111–112. Verlag der Technischen Universität Graz, 2017.
- [KS17c] Manuel Keglevic and Robert Sablatnig. Retrieval of striated toolmarks using convolutional neural networks. *IET Computer Vision*, 11(7):613–619, 2017.
- [KS18] Manuel Keglevic and Robert Sablatnig. Semi-Automatic Retrieval of Toolmark Images. In *Proceedings of the OAGM Workshop 2018*, pages 98–101. Verlag der Technischen Universität Graz, 2018.
- [KS19] Manuel Keglevic and Robert Sablatnig. Impress: Forensic Footwear Impression Retrieval. In *Proceedings of the ARW & OAGM Workshop 2019*, pages 167–169. Verlag der Technischen Universität Graz, 2019.
- [KSRF17] Bailey Kong, James Supancic, Deva Ramanan, and Charless Fowlkes. Fine-Grained Forensic Matching. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 188.1–188.12. BMVA Press, 2017.
- [KSRF19] Bailey Kong, James Supancic, Deva Ramanan, and Charless C. Fowlkes. Cross-Domain Image Matching with Deep Feature Maps. *International Journal of Computer Vision*, 127:1738–1750, 2019.
- [KV16] Adam Kortylewski and Thomas Vetter. Probabilistic Compositional Active Basis Models for Robust Pattern Recognition Probabilistic Compositional Active Basis Models for Robust Pattern Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 30.1–30.12. BMVA Press, 2016.
- [KWS19] Manuel Keglevic, Silvia Wilhelm, and Robert Sablatnig. Impress: A forensic footwear impression dataset. In *9th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pages 99–104. IET, 2019.
- [LBBH98] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [LBD<sup>+</sup>89] Yann LeCun, Bernhard Boser, John S. Denker, D. Michael Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 1989.
- [LBOM98] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Muller. Efficient BackProp. In *Neural Networks: Tricks of the Trade*, Springer Lecture Notes in Computer Sciences, pages 9–48. Springer, 1998.
- [LeC89] Yann LeCun. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.
- [LGSP13] Georgios Louloudis, Basilis Gatos, Nikolaos Stamatopoulos, and Aleksandros Papandreou. ICDAR 2013 Competition on Writer Identification. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1397–1401. IEEE, 2013.
- [LH05] Yann LeCun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 206–213. PMLR, 2005.
- [LL14] Tapio Luostarinen and Antti Lehmussola. Measuring the Accuracy of Automatic Shoeprint Recognition Methods. *Journal of Forensic Sciences*, 59(6):1627–1634, 2014.
- [LL19] Yuezun Li and Siwei Lyu. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 46–52. IEEE, 2019.
- [LWY<sup>+</sup>17] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 212–220. IEEE, 2017.
- [MAA<sup>+</sup>12] Sabri A. Mahmoud, Irfan Ahmad, Mohammad Alshayeb, Wasfi G. Al-Khatib, Mohammad Tanvir Parvez, Gernot A. Fink, Volker Märgner, and Haikal El Abed. Khatt: Arabic offline handwritten text database. In *2012 International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 449–454. IEEE, 2012.
- [MB02] Urs V. Marti and Horst Bunke. The IAM-database: An English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.
- [MBL20] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A Metric Learning Reality Check. In *European Conference on Computer Vision (ECCV)*, pages 681–699. Springer, 2020.

- [MJF<sup>+</sup>21] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image Matching from Handcrafted to Deep Features: A Survey. *International Journal of Computer Vision*, 129(1):23–79, 2021.
- [MMB01] Urs V. Marti, R. Messerli, and Horst Bunke. Writer Identification Using Text Line Based Features. In *2001 6th International Conference on Document Analysis and Recognition (ICDAR)*, pages 101–105. IEEE, 2001.
- [MP14] Naila Murray and Florent Perronnin. Generalized max pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.
- [MPT<sup>+</sup>17] John E. Murdock, Nicholas D. K. Petraco, John I. Thornton, Michael T. Neel, Todd J. Weller, Robert M. Thompson, James E. Hamby, and Eric R. Collins. The Development and Application of Random Match Probabilities to Firearm and Toolmark Identification. *Journal of Forensic Sciences*, 62(3):619–625, 2017.
- [MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [MTL<sup>+</sup>17] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No Fuss Distance Metric Learning Using Proxies. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 360–368. IEEE, 2017.
- [MWB<sup>+</sup>20] Erwin J. A. T. Mattijssen, Cilia L. M. Witteman, Charles E. H. Berger, Nicolaas W. Brand, and Reinoud D. Stoel. Validity and reliability of forensic firearm examiners. *Forensic Science International*, 307:110112, 2020.
- [NBC<sup>+</sup>09] Omar Nibouche, Ahmed Bouridane, Danny Crookes, Mourad Gueham, and Moussadek Laadjel. Rotation Invariant Matching of Partial Shoeprints. In *International Machine Vision and Image Processing Conference (IMVIP)*, pages 94–98. IEEE, 2009.
- [NBLK15] Angelos Nicolaou, Andrew D. Bagdanov, Marcus Liwicki, and Dimosthenis Karatzas. Sparse radial sampling LBP for writer identification. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 716–720. IEEE, 2015.
- [OSXJS16] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012. IEEE, 2016.
- [Ots79] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

- [PC13] Katerina S. Puentes and Hugo F. V. Cardoso. Reliability of cut mark analysis in human costal cartilage: The effects of blade penetration angle and intra- and inter-individual differences. *Forensic Science International*, 231(1-3):244–248, 2013.
- [PCDF<sup>+</sup>12] Nicholas D. K. Petraco, Helen Chan, Peter R. De Forest, Peter Diaczuk, Carol Gambino, James Hamby, Frani L. Kammerman, Brook W. Kammrath, Thomas A. Kubic, Loretta Kuo, Patrick McLaughlin, Gerard Petillo, Nicholas Petraco, Elizabeth W. Phelps, Peter A. Pizzola, Dale K. Purcell, and Peter Shenkin. Application of Machine Learning to Toolmarks - Statistically Based Methods for Impression Pattern Comparisons. Technical Report 239048, National Institute of Justice (NIJ), 2012.
- [PD07] Florent Perronnin and Christopher Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.
- [Pet11] Nicholas Petraco. *Color Atlas of Forensic Toolmark Identification*. CRC Press, 2011.
- [PSM10] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156. Springer, 2010.
- [PTB11] Mark Page, Jane Taylor, and Matt Blenkin. Uniqueness in the forensic identification sciences—Fact or fiction? *Forensic Science International*, 206(1):12–18, 2011.
- [PVZ15] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2015.
- [QSS<sup>+</sup>19] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 6450–6458. IEEE, 2019.
- [RB04] Nalini K. Ratha and Ruud Bolle, editors. *Automatic Fingerprint Recognition Systems*. Springer, New York, 2004.
- [RB21] Shervin Rasoulzadeh and Bagher BabaAli. Writer identification and writer retrieval based on NetVLAD with Re-ranking. *IET Biometrics*, 11(1):10–22, 2021.
- [RBCP19] Imad Rida, Sambit Bakshi, Xiaojun Chang, and Hugo Proenca. Forensic Shoe-print Identification: A Brief Survey. *CoRR*, abs/1901.01431, 2019.



- [RCLJ15] Joseph Roth, Andrew Carriveau, Xiaoming Liu, and Anil K. Jain. Learning-based Ballistic Breech Face Impression Image Matching. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2015.
- [RDS<sup>+</sup>15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Satheesh Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115:211–252, 2015.
- [RLL<sup>+</sup>17] Nicole Richetelli, Mackenzie C. Lee, Carleen A. Lasky, Madison E. Gump, and Jacqueline A. Speir. Classification of footwear outsole patterns using Fourier transform and local interest points. *Forensic Science International*, 275:102–109, 2017.
- [RMS<sup>+</sup>20] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 8242–8252. PMLR, 2020.
- [SCB07] H. Su, Danny Crookes, and Ahmed Bouridane. Shoeprint Image Retrieval by Topological and Pattern Spectra. In *International Machine Vision and Image Processing Conference (IMVIP)*, pages 15–22. IEEE, 2007.
- [SCBG07] H. Su, Danny Crookes, Ahmed Bouridane, and Mourad Gueham. Shoeprint Image Retrieval Based on Local Image Features. In *Proceedings of the Third International Symposium on Information Assurance and Security (IAS)*, pages 387–392. IEEE, 2007.
- [SCE<sup>+</sup>15a] Ryan Spotts, L. Scott Chumbley, Laura Ekstrand, Song Zhang, and James Kreiser. Angular Determination of Toolmarks Using a Computer-Generated Virtual Tool. *Journal of Forensic Sciences*, 60(4):878–884, 2015.
- [SCE<sup>+</sup>15b] Ryan Spotts, L. Scott Chumbley, Laura Ekstrand, Song Zhang, and James Kreiser. Optimization of a Statistical Algorithm for Objective Comparison of Toolmarks. *Journal of Forensic Sciences*, 60(2):303–314, 2015.
- [SHSP17] Johannes L. Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1482–1491. IEEE, 2017.
- [SKJJ11] Anuj Srivastava, Eric Klassen, Shantanu H. Joshi, and Ian Jermyn. Shape Analysis of Elastic Curves in Euclidean Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011.

- [SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. IEEE, 2015.
- [Soh16] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.
- [SRTV16] Joan A. Sánchez, Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 630–635. IEEE, 2016.
- [SSH21] Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 9367–9376. PMLR, 2021.
- [SV00] Jun-Feng Song and Theodore V. Vorburger. Proposed Bullet Signature Comparisons Autocorrelation Functions using. In *Proceedings of National Conference of Standards Laboratories*. NIST, 2000.
- [TFW17] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 661–669. IEEE, 2017.
- [TSKC11] Yi Tang, Sargur N. Srihari, Harish Kasiviswanathan, and Jason J. Corso. Footwear Print Retrieval System for Real Crime Scene Marks. In *International Workshop on Computational Forensics (IWCF 2010)*, pages 88–100. Springer, 2011.
- [TTY13] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue. A Survey of Keystroke Dynamics Biometrics. *The Scientific World Journal*, 2013:408280, 2013.
- [WB07] Simon A. J. Winder and Matthew Brown. Learning Local Image Descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.
- [WEY18] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. Improving generalization via scalable neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*, pages 685–701. Springer, 2018.
- [WHH<sup>+</sup>19] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-Similarity Loss With General Pair Weighting for Deep Metric

Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5022–5030. IEEE, 2019.

- [WMC21] Zhenghua Wang, Andreas Maier, and Vincent Christlein. Towards End-to-End Deep Learning-based Writer Identification. *INFORMATIK 2020*, pages 1345–1354, 2021.
- [WSL<sup>+</sup>14] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1393. IEEE, 2014.
- [WSYZ15] Xinnian Wang, Huihui Sun, Qing Yu, and Chi Zhang. Automatic Shoeprint Retrieval Algorithm for Real Crime Scenes. In Daniel Cremers, Ian Reid, Hideo Saito, and Ming-Hsuan Yang, editors, *Asian Conference on Computer Vision (ACCV 2014)*, pages 399–413. Springer, 2015.
- [WWZ<sup>+</sup>18] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274. IEEE, 2018.
- [WZW<sup>+</sup>17] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2593–2601. IEEE, 2017.
- [XQ16] Linjie Xing and Yu Qiao. DeepWriter: A Multi-stream Deep CNN for Text-Independent Writer Identification. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 584–589. IEEE, 2016.
- [YDT<sup>+</sup>19] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. Signal-to-noise ratio: A robust distance metric for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4815–4824. IEEE, 2019.
- [YT19] Baosheng Yu and Dacheng Tao. Deep metric learning with triplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 6490–6499. IEEE, 2019.
- [ZFD17] Yang Zhang, Huanzhang Fu, Emmanuel Dellandréa, and Liming Chen. Adapting Convolutional Neural Networks on the Shoeprint Retrieval for Forensic Use. In *Chinese Conference on Biometric Recognition (CCBR)*, pages 520–527. Springer, 2017.
- [ZK15] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361. IEEE, 2015.

- [ZL15] Jure Zbontar and Yann LeCun. Computing the Stereo Matching Cost With a Convolutional Neural Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1592–1599. IEEE, 2015.
- [ZW19] Andrew Zhai and Hao-Yu Wu. Classification is a Strong Baseline for Deep Metric Learning. In *Proceedings of the British Machine Vision Conference (BMVC)*, page 91. BMVA Press, 2019.
- [ZXD17] Hang Zhang, Jia Xue, and Kristin Dana. Deep TEN: Texture Encoding Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2896–2905. IEEE, 2017.
- [ZZCL17] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking Person Re-identification with k-reciprocal Encoding. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3652–3661. IEEE, 2017.