UNIVERSITY OF CALIFORNIA,
IRVINE

Computational Shoeprint Analysis for Forensic Science

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Samia Shafique

Dissertation Committee:
Professor Charless C. Fowlkes, Chair
Associate Professor Alexander C. Berg
Professor Erik B. Sudderth

2024

DEDICATION

To my parents who have supported and encouraged me all my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Charless Fowlkes, for his help with my research. He has provided me with excellent guidance and has been very kind and supportive throughout my PhD. The completion of this dissertation would not have been possible without his support.

I am very grateful to my final defense committee members: Professor Charless Fowlkes, Professor Alexander Berg, and Professor Erik Sudderth; and my additional advancement committee members: Professor Shuang Zhao, Professor Aditi Majumder, and Professor Hal S. Stern.

I am fortunate to have been part of the Computational Vision Lab at UCI, where I met wonderful peers who provided a stimulating environment and helpful discussions: Shu Kong, Bailey Kong, Minhaeng Lee, Daeyun Shin, Phuc Nguyen, and Zhe Wang. I am very grateful to Shu Kong for being my co-author and helping me structure my research.

Furthermore, I would like to extend my gratitude to Professor Aditi Majumder and Professor Gopi Meenakshisundaram for their support and guidance in the early years of my PhD. I would also like to thank my friends from the iGravi Lab, who became a second family to me when I moved thousands of miles from home to start my PhD at UCI: Zahra Montazeri, Yu Guo, Marco (Zhanhang) Liang, Cheng Zhang, Muhammad Twaha Ibrahim, and Nitin Agarwal.

I would like to thank my family for their understanding and unwavering love. My parents, Muhammad Shafiqul Islam and Shahnaz Begum, have patiently raised me, sometimes sacrificing their own dreams to see mine come true. My wonderful husband, Nafiul Rashid, has been by my side, helping me remain steadfast throughout this journey. My adorable son, Zayd Rashid, has been my source of encouragement on gloomy days.

Finally, I would like to thank the Center for Statistics and Applications in Forensic Evidence (CSAFE) for supporting my research with grants through Cooperative Agreements 70NANB15H176 and 70NANB20H019.

VITA

Samia Shafique

EDUCATION

Doctor of Philosophy in Computer Science — **2024**
University of California, Irvine — Irvine, California, USA

Master of Science in Computer Science — **2019**
University of California, Irvine — Irvine, California, USA

Bachelor of Science in Computer Science and Engineering — **2016**
Bangladesh University of Engineering and Technology — Dhaka, Bangladesh

RESEARCH AND INDUSTRY EXPERIENCE

Graduate Research Assistant — **2019–2024**
University of California, Irvine — Irvine, California, USA

ML Software Engineering Intern — **2023**
Uber — Sunnyvale, California, USA

Platform Engineering Intern — **2018**
Supplyframe — Pasadena, California, USA

TEACHING EXPERIENCE

Teaching Assistant — **2017–2019**
University of California, Irvine — Irvine, California, USA

Lecturer — **2016–2017**
United International University — Dhaka, Bangladesh

REFEREED PUBLICATIONS

CriSp: Leveraging Tread Depth Maps for Enhanced Crime-Scene Shoeprint Matching **Sep 2024**
S Shafique, S Kong, C Fowlkes. Under review at European Conference on Computer Vision (ECCV)

Creating a Forensic Database of Shoeprints from Online Shoe-Tread Photos **Jan 2023**
S Shafique, B Kong, S Kong, C Fowlkes. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

SOFTWARE

Shoe Retrieval Tool https://github.com/Samia067/crisp_app
A tool to retrieve shoe models that most closely match a query crime-scene shoeprint.

ABSTRACT OF THE DISSERTATION

Computational Shoeprint Analysis for Forensic Science

By

Samia Shafique

Doctor of Philosophy in Computer Science

University of California, Irvine, 2024

Professor Charless C. Fowlkes, Chair

Shoeprints are a common type of evidence found at crime scenes and are regularly used in forensic investigations. However, their utility is limited by the lack of reference footwear databases that cover the large and growing number of distinct shoe models. Additionally, existing methods for matching crime-scene shoeprints to reference databases cannot effectively employ deep learning techniques due to a lack of training data. Moreover, these methods typically rely on comparing crime-scene shoeprints with clean reference prints instead of more detailed tread depth maps. To address these challenges, we break down the problem into two parts. First, we leverage shoe tread images sourced from online retailers to predict their corresponding depth maps, which are then thresholded to generate prints, thus constructing a comprehensive reference database. Next, we use a section of this database to train a retrieval network that matches query crime-scene shoeprints to tread depth maps. Extensive experimentation across multiple datasets demonstrates the state-of-the-art performance achieved by both the database creation and retrieval steps, validating the effectiveness of our proposed methodology.

Chapter 1

Introduction

Examining evidence from a crime scene helps investigators identify suspects. Shoeprints are often found at crime scenes, though they may have fewer distinct identifying features compared to other biometric samples like blood or hair [16]. Nonetheless, analyzing shoeprints can provide crucial leads that help investigators narrow down potential suspects.

In recent years, significant progress has been made in automated shoeprint matching. Some works have introduced databases of crime-scene impressions and a corresponding set of reference lab impressions [50, 89], while others have proposed methods to generate features and find similarities between crime-scene shoeprints and reference databases [48, 47, 98, 76, 60]. Unfortunately, current methods suffer from several fundamental limitations. First, the reference databases proposed have been manually curated, making it impractically tedious to maintain a comprehensive, large-scale database. Second, the matching algorithms are not generalizable because they largely rely on hand-crafted or pretrained network features.

To overcome these limitations, we propose the automated creation of a large-scale reference database by crawling shoe tread images advertised by online retail shops and generating depth map and

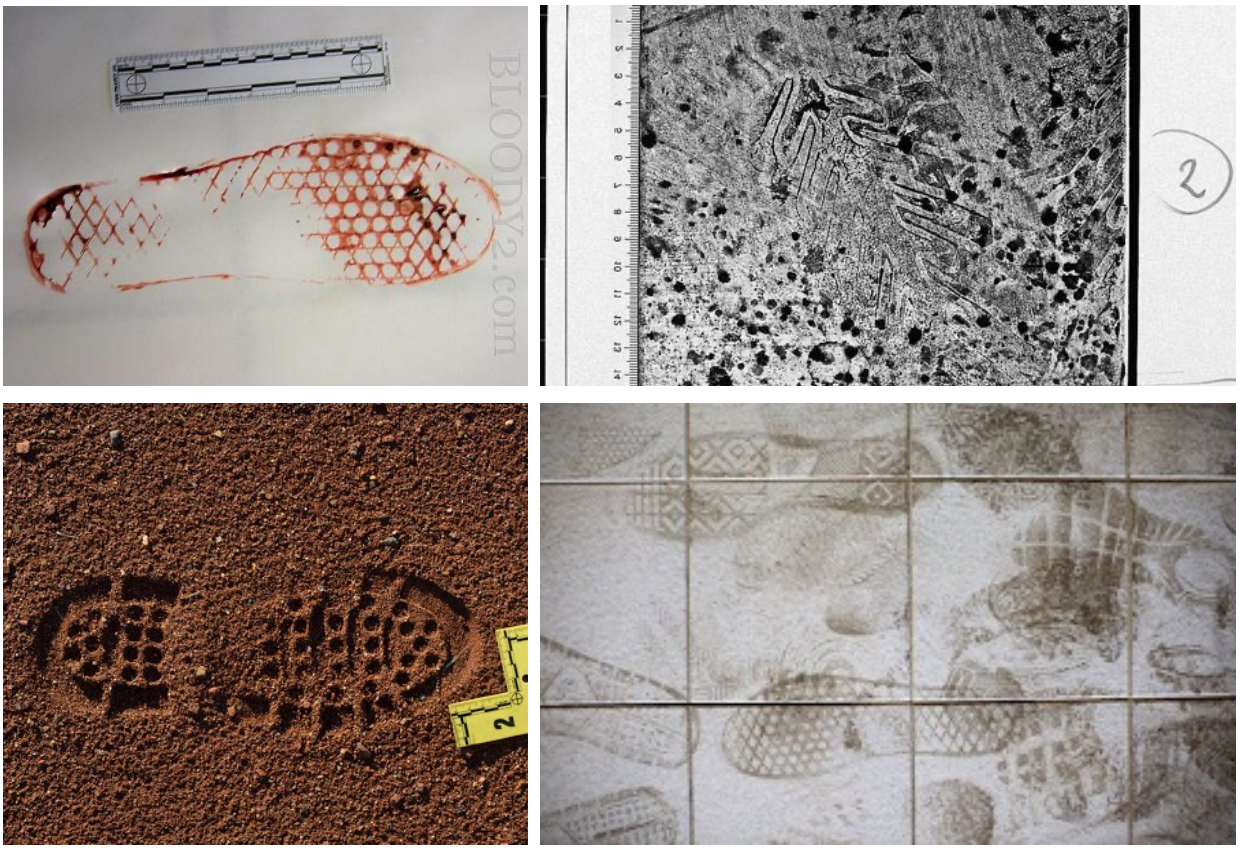

Figure 1.1: Crime-scene shoeprints are typically very noisy and appear across various mediums, as shown in the images above. They are often partially visible and significantly degraded, with clarity frequently compromised by overlapping prints. These prints can vary in type, such as those made by dust or blood, and can occur on hard or soft surfaces. Therefore, methods for retrieving the closest matching shoe models must be capable of handling these complex cases.

shoeprint predictions from them. Subsequently, we use a portion of this data to train a retrieval network for matching crime-scene shoeprints to tread depth maps.

In this chapter, we start with a detailed description of the problem and its significance in Section 1.1. Section 1.2 provides an overview of the general framework for automated shoeprint matching systems. Next, we summarize the contributions of this dissertation in Section 1.3 and acknowledge funding sources in Section 1.4. Section 1.5 provides an outline of its organization.

Figure 1.2: Our goal is to identify the class characteristics of crime-scene shoeprints rather than their acquired characteristics. Class attributes encompass general features of the shoe, such as brand, model, and size. Acquired attributes are the unique traits that develop on a shoe over time, such as holes, cuts, and scratches.

1.1 Problem Definition and Motivation

The goal of this dissertation is to assist investigators in the forensic analysis of footwear evidence left at crime scenes by generating a list of shoe models that best match a query shoeprint. Specifically, we aim to provide investigators with a ranked list of potential matches to a crime scene shoeprint, even when it is severely degraded or only partially visible. Figure 1.1 shows example crime-scene shoeprints, illustrating the challenges in analyzing these images. Shoeprints can be made by various mediums (blood, dust, etc.) on different surfaces (hard tiles, soft sand, etc.), resulting in prints with diverse characteristics. They can be occluded by overlapping prints or other marks, and may only be partially visible. Retrieval methods must be capable of handling these complexities.

The forensic examination of shoeprints provides insights into both the class attributes and the acquired attributes of the perpetrator's footwear. Class attributes include general features of the

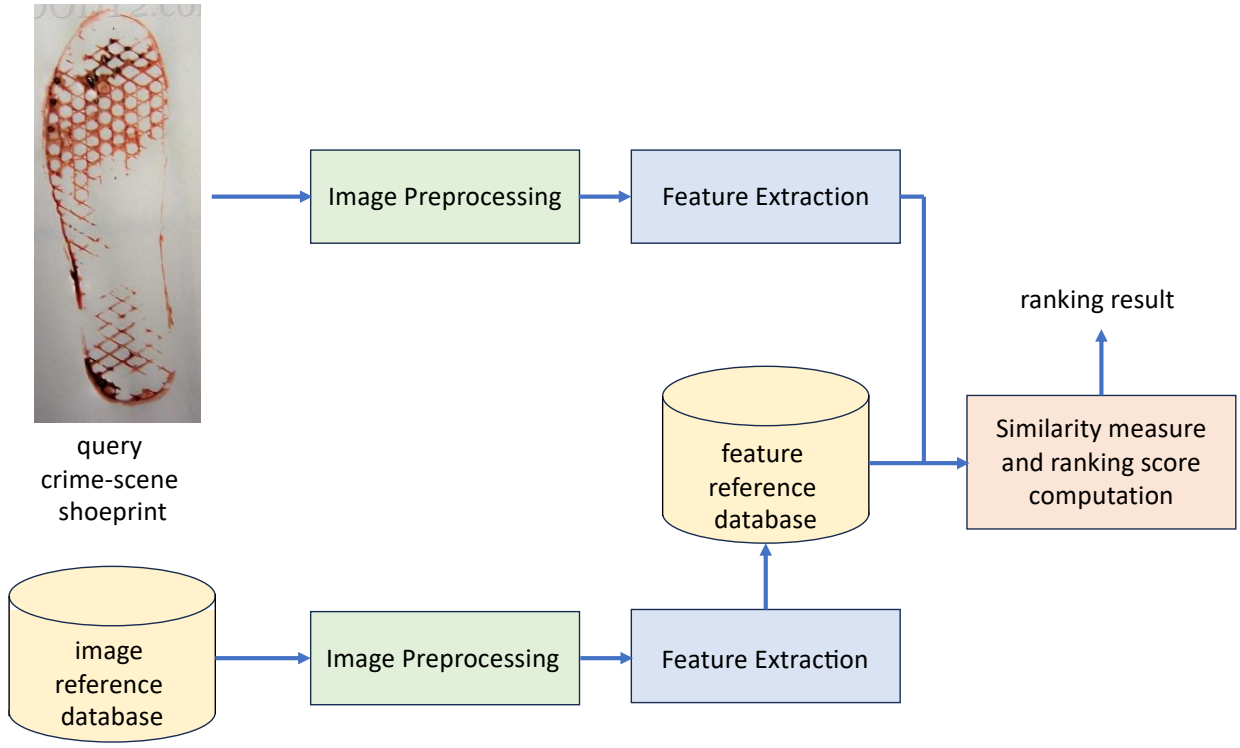

Figure 1.3: General framework for automated shoeprint matching systems. Clean shoeprint images from a reference database are preprocessed and features are extracted from it. These database features are stored, and whenever a new query image appears, it is also processed to generate features and these features are compared to those from the database images to generate a ranking of shoe models.

shoe, such as brand, model, and size. Acquired attributes refer to the unique characteristics that develop on a shoe over time, such as holes, cuts, and scratches. Our focus is on facilitating the investigation of the class attributes of shoeprints. This is illustrated in Figure 1.2.

1.2 Framework For Automated Shoeprint Matching Systems

In this dissertation, we investigate the automated matching of crime-scene shoeprints to aid forensic analysis. An overview of automated shoeprint matching systems is shown in Figure 1.3. A query crime-scene shoeprint undergoes preprocessing and feature extraction. Likewise, clean shoeprint images from a reference database are processed to extract their features. These features

are stored in the database, and when a new query image is introduced, its features are compared to the stored features to produce a ranking of shoe models. A forensic investigator can be expected to look through the retrieved shoe models and make a judgment of whether the retrievals are true matches to the crime-scene shoeprint.

1.3 Our Contributions

We make contributions in two parts of the automated shoeprint matching framework.

- Reference database. Currently, the proposed reference databases have all been manually curated, making their maintenance expensive and requiring tremendous human effort. To address this, we introduce a method to automatically generate a large-scale reference database that encompasses the vast and growing number of shoe tread patterns [81]. Our approach involves crawling online retail stores that advertise shoe tread images and predicting the depth maps and shoeprint patterns from these images. We develop a method called *ShoeRinsics* that incorporates intrinsic image decomposition and domain adaptation techniques, outperforming prior art for this task. We develop a benchmarking protocol, with which we evaluate existing methods of depth prediction using domain adaptation for this task.

- Feature extraction. While current literature relies on hand-crafted features or features from pre-trained networks, we are the first to train a model on a large-scale database to extract generalizable and relevant features [82]. Additionally, we introduce the concept of matching crime-scene shoeprints to tread depth maps in the reference database instead of shoeprints, leading to improved retrieval performance compared to traditional shoeprint matching methods. We develop a spatially-aware matching method named *CriSp*, which demonstrates superior performance over existing methods in both shoeprint matching and image retrieval tailored to this specific task. We propose a benchmarking protocol to evaluate our method

against state-of-the-art approaches. This involves the creation of a new dataset, reprocessing of existing datasets, and defining appropriate evaluation metrics.

1.4 Acknowledgements

This work was funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreements, 70NANB15H176 and 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

1.5 Dissertation Outline

The rest of this dissertation is organized as follows. In Chapter 2, we introduce a method to generate a reference database of tread images by crawling images from online retail stores, predicting their depth maps, and thresholding these depth maps to make print predictions. In Chapter 3, we use data generated in this manner to train a retrieval network to match crime-scene shoeprints to a reference database of tread depth maps. Chapter 4 concludes the dissertation with a discussion on limitations and future directions.

Chapter 2

Creating a Forensic Database of Shoeprints from Online Shoe-Tread Photos

2.1 Introduction

Studying the evidence left at a crime scene aids investigators in identifying criminals. Shoeprints have a greater chance of being present at crime scenes [16], although they may have fewer uniquely identifying characteristics than other biometric samples (such as blood or hair). Thus, studying shoeprints can provide valuable clues to help investigators narrow down suspects of a crime.

Forensic analysis of shoeprints can provide clues on the *class characteristics* and the *acquired characteristics* of the suspect's shoe. The former involves the type of shoe (e.g., the brand, model, and size); the latter consists of the individual traits of a particular shoe that appear over time as it is worn (e.g., holes, cuts, and scratches). We are interested in aiding the study of class characteristics of shoeprints.

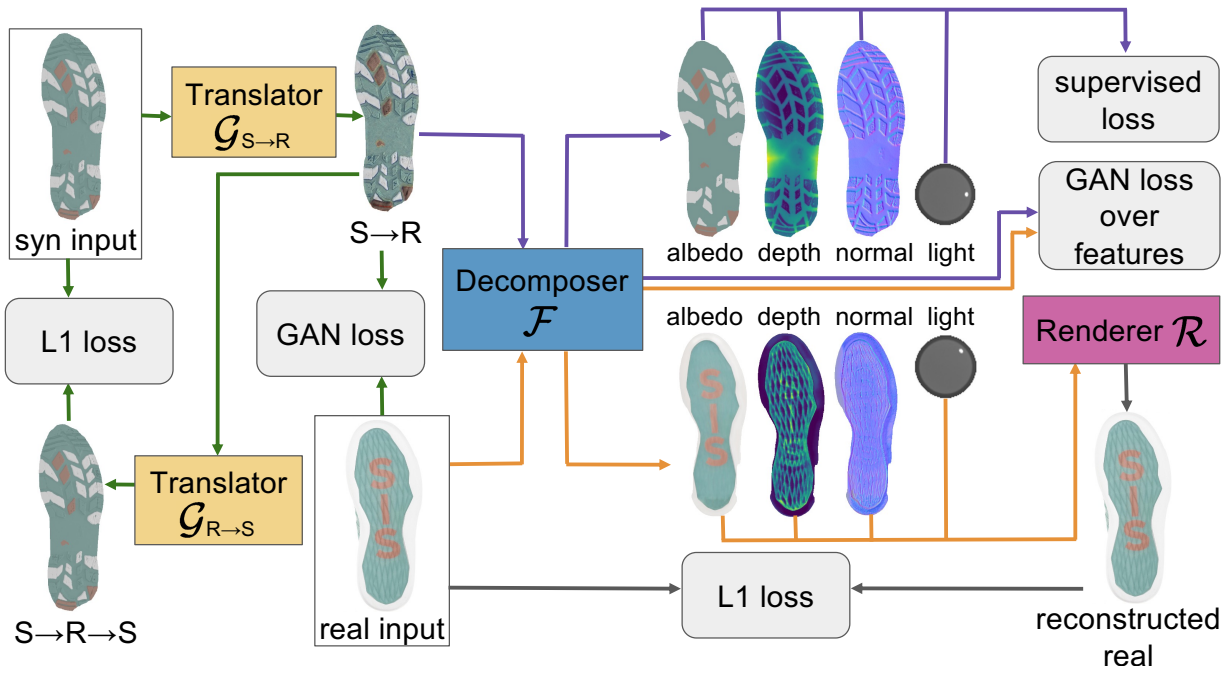

Figure 2.1: Predicting depth for shoe-tread images (collected by online retailers) is the core challenge in constructing a shoeprint database for forensic use. We develop a method termed *ShoeRinsics* to learn depth predictors. The flowchart depicts how we train our *ShoeRinsics* using annotated synthetic and un-annotated real images (Sec. 2.4). We use domain adaptation (via image translators $\mathcal{G}_{S\to R}$ and $\mathcal{G}_{R\to S}$) and intrinsic image decomposition (via decomposer $\mathcal{F}$ and renderer $\mathcal{R}$) techniques to mitigate synthetic-real domain gaps (Sec. 3.5). Our method achieves significantly better depth prediction on real shoe-tread images than the prior art (Sec. 2.6).

Status quo. Traditionally, investigating class characteristics of shoeprints involve matching the prints against a *manually curated* database of impressions of various shoe models [18]. The research community has shown significant interest in automating this matching process [17, 22, 36, 37, 38, 5, 45, 52, 48, 95, 107]. However, in practice, the success of such work depends on the quality of the database to which the shoeprint evidence is compared. Yet, maintaining and regularly updating such a database to include all shoe models is tedious, costly, and requires significant human effort. Shoeprint matching methods are decidedly less useful if the database does not include the type of shoe the criminal wore! Partly because of this, shoeprint evidence is vastly underutilized in the USA [85].

Motivation. To address the need for such a comprehensive database, we propose to leverage imagery of shoe-treads collected by online retailers. High-resolution tread photos of various shoe

products are readily available, and shopping websites are updated frequently (>1000 new products appear each month based on our analysis on some websites). Fig. 2.2 (b) shows examples of such shoe-tread images. Developing a method to predict the 3D shape from a shoe-tread image would directly address the need for a comprehensive, up-to-date database of tread patterns. *We formulate this problem as depth prediction for shoe-treads; thresholding the depth map of a given shoe can generate/simulate shoeprints sufficient for matching query prints.*

Technical Insights. To learn depth predictors from single shoe-tread images, we would ideally utilize supervised training examples of aligned shoe-tread images and their corresponding depth maps. However, since such ground-truth data is simply unavailable, we develop an alternative strategy. We create a synthetic dataset of rendered shoe-tread images and corresponding ground-truth depth, albedo, normal, and lighting. This data can train a predictor in a fully supervised fashion. However, the resulting model performs sub-optimally on real-world images due to the domain gap between synthetic and real imagery. To address this, we introduce three additional techniques to close the synthetic-real domain gap by incorporating methods of domain adaptation [113] and intrinsic image decomposition [44] (see Fig. 2.1). First, we train a translator that translates synthetic shoe-treads to realistic images, which better match the distribution of the real shoe-treads. Second, we use an adversarial loss to enforce that features of real and translated synthetic images are indistinguishable. Third, we use a re-rendering loss that adopts a synthetically trained renderer to reconstruct the real shoe-tread images using their predicted depth and other intrinsic components. We find these three techniques in combination help close the domain gap and yield significantly better depth prediction.

Contributions. We make three major contributions.

- Motivated to create a database of shoeprints for forensic use, we introduce the task of depth prediction for real shoe-tread photos collected by online retailers.

- We develop a benchmarking protocol, with which we evaluate existing methods of depth prediction using domain adaptation for this task.
- We develop a method called *ShoeRinsics* that incorporates intrinsic image decomposition and domain adaptation techniques, outperforming prior art for this task.

2.2 Related Work

Shoeprint Analysis. Automatic shoeprint matching has been studied widely in the past two decades [74]. Existing works focus on generating good features from shoeprints and using them to assign a class label (shoe type) from a database of lab footwear impressions. To study global features (i.e., considering the whole shoe), [52] introduces a probabilistic compositional active basis model, [48] explores multi-channel normalized cross-correlation to match multi-channel deep features, and [95] employs a manifold ranking method, and [107] uses VGG16 as a feature extractor. On the other hand, [63] studies a multi-part weighted CNN, [7] introduces a block sparse representation technique, and [8] applies multiple point-of-interest detectors and SIFT descriptors to study the local features of shoeprints (i.e., keypoints [53]). Our work differs from the previous work as it focuses on creating a database of prints rather than developing methods for shoeprint matching. *Creating such as database is a prerequisite for algorithmic explorations for shoe-matching.*

Monocular Depth Prediction has been studied extensively since early works [42, 78, 77]. Previous methods invent features representations [13, 73, 32], deep network architectures [9, 57, 75, 46, 56, 101], and training losses [30, 84, 104]. [55, 34, 59] explore self-supervised learning in a stereo setup while [72, 109] experiment with training on large datasets. Depth estimation has been further improved by considering the camera pose [108]. Our work differs from the above as it aims for depth prediction on real images by learning over un-annotated real images and synthetic images (and their ground-truth intrinsics: depth, albedo, normal and light).

Intrinsic Image Decomposition. Another line of work aims to explain image appearance in terms of some intrinsic components, including albedo, normals, and lighting. However, predicting intrinsic images is difficult, if not impossible. Our approach is related to [44], which learns for intrinsic image decomposition and uses a differentiable renderer to leverage un-annotated images with a reconstruction loss. [80, 64, 96] focus on face images and explore a similar reconstruction loop [80], non-diffuse lighting models [64], and multiple reflectance channels [96]. [99] works on rotationally symmetric objects with only object silhouettes as supervision. [79, 106, 112, 58, 105] study decomposition on entire scenes. [6] learns photo-realistic rendering of synthetic data and intrinsic decomposition of real images using unpaired data as input via an adversarial loss. In contrast, our work utilizes intrinsic decomposition techniques to help learn depth prediction by leveraging annotated synthetic and un-annotated real data via domain adaptation.

Domain Adaptation. Training solely on synthetic data can cause models to perform poorly on real data. Adversarial domain adaptation has proved promising for bridging such domain gaps. One way to approach this is to use domain-invariant features to map between the domains. [61] proposes to reduce the Maximum Mean Discrepancy to learn domain-invariant features. [94] builds on this idea and further improves domain adaptation performance in classification tasks. [93, 92, 33, 88] learn domain adaptation by aligning source and target features. Another direction of work uses image-to-image translation [113] to stylize source images as target images. [41, 110] use the stylized source images to learn from target images using source labels while performing alignment both at the image and feature level. We use domain adaptation for depth estimation but take this approach further by reasoning about the intrinsic components of unlabeled real data.

Table 2.1: Overview of our datasets for training and testing, along with their shoe categories and counts. It is worth noting that real-val contains formal and used shoes, which are not present in training (i.e., the real-train set). We include these novel shoe types to analyze the generalizability of different methods. See details in Sec. 2.4 and visual examples in Fig. 2.2.

Dataset	Shoe Category			Total	Annotation
	New-Athletic	Formal	Used		
syn-train	88,408	0	0	88,408	depth, albedo, normal, light
real-train	3,543	0	0	3,543	none
real-val	22	6	8	36	print
real-FID-val	41	0	0	41	print

2.3 Problem Setup and Evaluation Protocol

Our motivation is to create a database of shoeprints for forensic use. *The specific task is to predict depth maps for shoe-tread images collected by online retailers.* Below, we formulate the problem and introduce an evaluation protocol to benchmark methods.

2.3.1 Problem Setup

Online shoe-tread photos do not have ground-truth depth. Thus, we cannot directly train a depth predictor on them. Instead, we propose to create a dataset of synthetic shoe-tread images for which we have a complete set of annotations, including depth, albedo, normal, and lighting (details in Section 2.4.1). Therefore, **the problem is to predict depth for *real* shoe-treads by learning a depth predictor on synthetic shoe-treads (with annotations) and real shoe-treads (without annotations)**. This requires (1) learning a depth predictor by exploiting synthetic data that has annotations of depth and other intrinsic components, (2) addressing the synthetic-real domain gap.

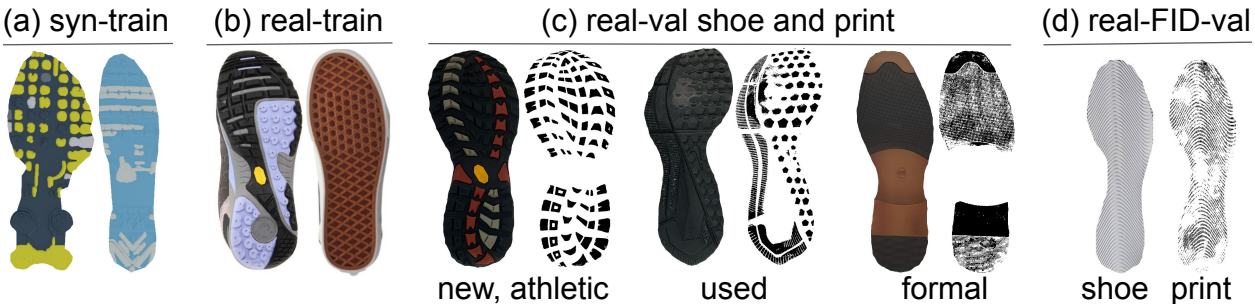

Figure 2.2: Shoe tread examples from (a) syn-train, (b) real-train, (c) real-val, and (d) real-FID-val. Clearly, a domain gap exists between (a) syn-train and (b) real-train, demonstrating the need to close the synthetic-real domain gap. Moreover, to study the generalizability, we evaluate on 2 datasets (c) and (d) and purposely hold out the formal and used shoe-treads which are not used for training but for validation (c).

2.3.2 Evaluation Protocol

Recall that the created database, containing predicted depth maps and shoe-tread images, and will serve for forensic use – an investigator will query a shoeprint collected at a crime scene by matching it with depth maps within this database. Therefore, we evaluate the quality of predicted depth maps w.r.t shoeprint matching.

To this end, we introduce two validation sets that contains paired "ground-truth" shoeprints and shoe-tread photos (details in Section 2.4.2). For a given shoe-tread, a trained model predicts its depth and the metric measures the degree of match between the ground-truth shoeprint and the predicted depth. We develop a metric based on Intersection-over-Union (IoU). Specifically, we generate a set of shoeprints using adaptive thresholding (with a range of hyperparameters) for the predicted depth, and compute the IoU between the ground-truth print to each of these generated shoeprints. The metric returns the highest IoU. We further average the IoUs over all the validation data as mean IoU (mIoU) to benchmark methods. Refer to the supplement for further details.

2.4 Data Preparation

During training, we have two data sources: a synthetic dataset (*syn-train*) that has annotations, and a dataset of un-annotated real shoe-treads (*real-train*). To study models' generalizability, we test our model on two validation sets (*real-val* and *real-FID-val*). Each of these datasets contain shoe-tread photos with aligned ground-truth shoeprints, which enable quantitative evaluation. Note that to analyze the models' robustness to novel shoe types, we constrain our training sets to contain only brand-new athletic shoes while letting real-val also include formal and used (worn) shoes. Fig. 2.2 displays example shoe-treads and Table 2.1 summarizes the four datasets. Below, we elaborate on the creation of the synthetic training set (syn-train), the real training set (real-train), and validation sets (real-val and real-FID-val).

2.4.1 Synthetic Data for Training

Our synthetic dataset (syn-train) containing synthetic shoe-tread images and their intrinsic annotations (depth, albedo, normal, and lighting). We synthesize a shoe-tread image with a given depth map, an albedo map, and a lighting environment (outlined in Fig. 2.3). We pass these to a physically-based rendering engine [43] to generate the synthetic image. The final syn-train set contains 88,408 shoe-treads with paired ground-truth intrinsic images.

Depth Map. We use an existing dataset [102] to generate plausible synthetic depth maps to create syn-train data. For each of 387 shoeprints, we synthesize 10-15 different depth maps. Because the shoeprints have noise that affects synthetic data generation, we first apply a Gaussian blur to filter the noise. We then scale the blurred print image to create a "pseudo" depth map. To generate more diverse depth maps we add random high-frequency textures. Lastly, we make tread shapes more realistic by adding a priori features, such as slanted bevels on the tread elements and global curvature of the shoe-tread (details in supplement).

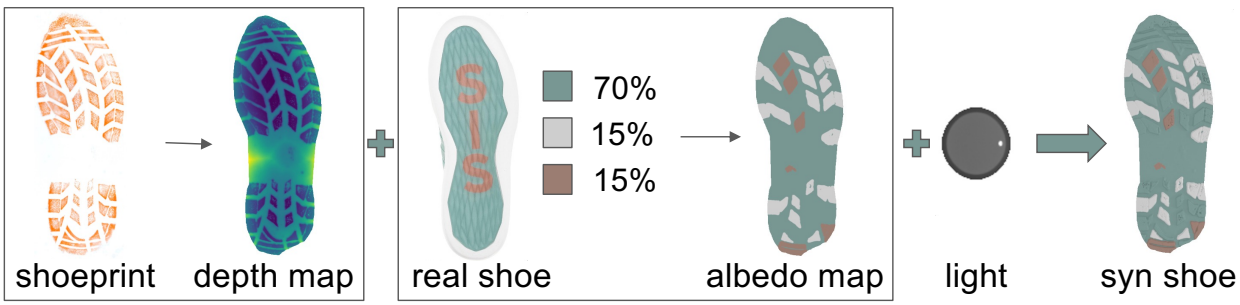

Figure 2.3: Generation of synthetic data. We scale off-the-shelf shoeprints to generate "pseudo" depth maps. We sample a color distribution from a real-shoe example to create an albedo map. The depth map and albedo map are combined with a lighting environment to render a synthetic image. The lighting environment is demonstrated by visualizing a shiny sphere in place of the shoe. In this example, directional light comes from a point on the right.

Albedo Map. The color palette for each rendered shoe comes from the color distribution of a real shoe-tread photograph. Shoes tend to have only a handful of different colors across the entire tread. We identify the primary colors on real shoe-treads using the mean-shift algorithm [31]. Albedo maps for the rendered shoes are composed of these colors. First, we use depth maps to identify shoe-tread elements and segment out areas of the shoe that can have different colors. Then we assign colors to those segments from the color palette of a real shoe in the percentages in which they are present. Fig. 2.3 shows one example.

Light environment. Online retail stores use specialized diffuse lighting rigs to capture shoe photos. We create a similar lighting environment for our rendered images. Shoes are photographed with bright diffuse white light from all directions and some optional directional light. We use a total of 17 different light configurations. One light configuration is simply diffuse light coming from all directions. Eight light configurations consist of single light bulbs shining from eight directions around the shoe in addition to the diffuse white light. The remaining eight are similar but contains two light bulbs at 120° to each other. The supplement has further details.

Figure 2.4: Generating pseudo albedo maps from shoe-tread images. We show two pairs. We run the mean-shift algorithm [31] on a shoe-tread image to group RGB pixels, resulting in the corresponding pseudo albedo map. We use the pseudo albedo maps as supervision signals to train the decomposer (cf. Fig. 2.1).

2.4.2 Online Shoe Treads for Training and Prediction

Online retailers [1, 3] adopt photos of shoes for advertisement, which include shoe-tread images. Real-train (3,543), cf. Table 2.1, consists of such shoe-tread images and masks computed by a simple network to segment out the shoe-treads. This dataset does not contain any ground-truth and consists only of new, athletic shoes.

2.4.3 Lab Data for Validation

Real-val. To quantitatively benchmark methods, we collect paired shoe-tread images and ground-truth prints in a lab environment. Fig. 2.5 summarizes the procedure. We photograph shoes by placing them inside a light box with a ring light on top. We collect prints from those shoes by painting the treads with a thin layer of relief ink and pressing absorbent white papers onto the shoe-treads. This method of collecting shoeprints is called the *block printing technique* and is one of several techniques used in the forensics community to collect reference footwear impressions [16]. To improve print quality, we collect 2-3 prints for each shoe and average them after alignment to the shoe-tread. We use thin-plate splines [25] with a smoothness parameter of 0.5 for alignment. We threshold the average print as the final ground-truth shoeprint. Real-val contains 22 new-athletic shoes, 6 new formal shoes, and 8 used athletic shoes. The formal and used shoes are not present during training and thus serve as novel examples in evaluation.

Real-FID-val. We introduce the second validation set consisting of shoeprints from the FID300 dataset [50] and shoe-tread images separately downloaded from online retailers (i.e., these images are disjoint from those in the real-train set). We find matched FID300 prints (used as the ground-truth) and the downloaded shoe-tread images, and align them manually. Real-FID-val contains 41 new, athletic shoe-tread images with corresponding ground-truth shoeprints and masks to segment out the shoe-treads.

2.5 Methodology

We now introduce our *ShoeRensics*, a pipeline that trains a depth predictor for real images I_R by incorporating unsupervised adversarial domain adaptation and intrinsic image decomposition techniques. Given synthetic images I_S with their corresponding ground-truth intrinsics (albedo X_S^a, depth X_S^d, normal X_S^n, and light X_S^l) and unlabeled real images I_R, our goal is to train a model to predict depth d_R for real images I_R. Fig. 2.1 overviews our training pipeline. The main components of our pipeline are a translator $\mathcal{G}_{S\to R}$ to stylize synthetic images as real images, a decomposer $\mathcal{F}$ for intrinsic image decomposition, and a renderer $\mathcal{R}$ to reconstruct the input images from their intrinsic components.

Synthetic-only Training. We train a decomposer $\mathcal{F}$ and the renderer $\mathcal{R}$ in a supervised manner on syn-train. For an input image, the decomposer predicts depth $\hat{X}_S^d$, albedo $\hat{X}_S^a$, normal $\hat{X}_S^n$, and light $\hat{X}_S^l$. The renderer $\mathcal{R}$ learns to reconstruct the input image from these predicted intrinsic components. To train the decomposer $\mathcal{F}$, we use an $\mathcal{L}_1$ loss to learn for depth, albedo, and normal prediction and a cross-entropy loss $\mathcal{L}_{CE}$ to learn for light (treating light prediction as a K-way classification problem given the limited light sources). We minimize the overall loss below:

$$\mathcal{L}_{sup} = \lambda_l \mathcal{L}_{CE}(\hat{X}_S^l, X_S^l) + \sum_{\kappa\in\{d,a,n\}} \lambda_\kappa \mathcal{L}_1(\hat{X}_S^\kappa, X_S^\kappa). \tag{2.1}$$

Figure 2.5: We collect a validation set of ground-truth shoeprints from shoes in a lab environment. (a) shows an example shoe. (b) It is painted with a thin layer of relief ink, and a paper sheet is pressed evenly onto the shoe-tread using a roller. (c) We repeat this to get 2-3 different prints. (d) We align these prints to the shoe-tread using thin-plate spline [25] and (e) threshold their average to obtain the final ground-truth shoeprint, which has better coverage.

where λ's are hyperparamters controlling loss terms for the intrinsic components. To learn the renderer $\mathcal{R}$, we simply minimize the $\mathcal{L}_1$ loss between the original and rendered images, i.e., $\mathcal{L}_1(I_S, \mathcal{R}(X_S^d, X_S^a, X_S^n, X_S^l))$. Note that depth prediction is our main focus, and we find learning with decomposer and renderer significantly helps depth learning (cf. Fig. 2.1, Table 2.2). A model trained on synthetic data only does not work effectively well on real data due to the notorious synthetic-real domain gap. We address this issue using the techniques below.

Mitigating domain gap by image translation. Previous work [41, 113] addresses the domain gap between image sources by translating images from one domain to the other. We adopt a similar approach and translate our synthetic images to realistic ones by training a translator $\mathcal{G}_{S\to R}$. We train another $\mathcal{G}_{R\to S}$ that translates real images to synthetic style. Discriminators $\mathcal{D}_R(I)$ and $\mathcal{D}_S(I)$ are learned simultaneously to discriminate translated images and used for training translators. This is known as the adversarial domain adaptation [41]. We further translate the translated synthetic/real images back to the original domain and use a cycle loss between the resulting and the initial images to ensure that structure and content are preserved during translation. The following losses train the

translators [41, 113]:

$$\mathcal{L}_{GAN}^{S\to R}(I_R, I_S) = \log \mathcal{D}_R(I_R) + \log(1 - \mathcal{D}_R(\mathcal{G}_{S\to R}(I_S)))$$
$$\mathcal{L}_{GAN}^{R\to S}(I_S, I_R) = \log \mathcal{D}_S(I_S) + \log(1 - \mathcal{D}_S(\mathcal{G}_{R\to S}(I_R)))$$
$$\mathcal{L}_{tran} = \mathcal{L}_{GAN}^{S\to R}(I_R, I_S) + \mathcal{L}_{GAN}^{R\to S}(I_S, I_R)$$
$$\mathcal{L}_{cyc} = \mathcal{L}_1(\mathcal{G}_{R\to S}(\mathcal{G}_{S\to R}(I_S)), I_S) +$$
$$\mathcal{L}_1(\mathcal{G}_{S\to R}(\mathcal{G}_{R\to S}(I_R)), I_R) \tag{2.2}$$

With $\mathcal{G}_{S\to R}(I_S)$, we translate syn-train images and keep their corresponding ground-truth intrinsics unchanged. We use such translated data to finetune the renderer $\mathcal{R}$.

Mitigating domain gap by image reconstruction. We additionally use an image reconstruction loss to address the domain gap [44]. We reconstruct a real image from its decomposed intrinsic components using the trained renderer $\mathcal{R}$, which we freeze after finetuning on translated synthetic data. We use $\mathcal{R}$ to regularize the training of the decomposer $\mathcal{D}$ on real images. Denoting reconstructed real image as $\hat{I}_R \coloneqq \mathcal{R}(X_R^d, X_R^a, X_R^n, X_R^l)$, we minimize the difference between the original image I_R and its reconstruction $\hat{I}_R$ using an $\mathcal{L}_1$ loss, i.e., $\mathcal{L}_1(\hat{I}_R, I_R)$.

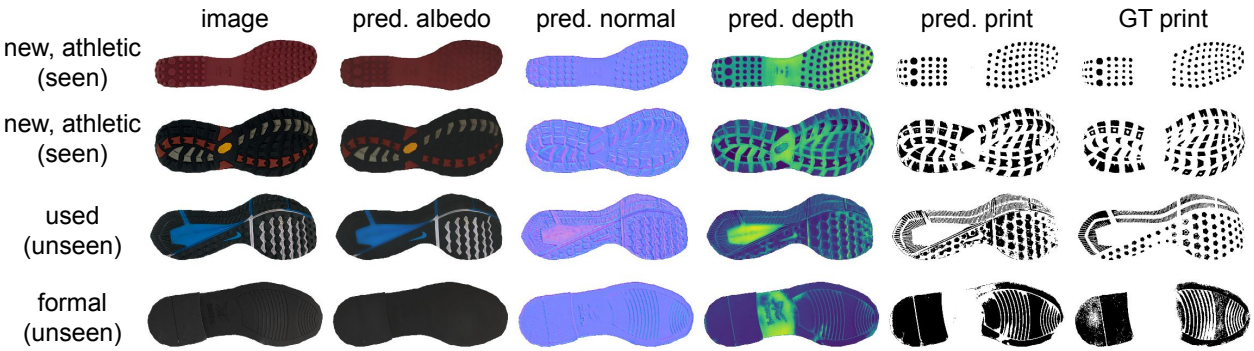

Figure 2.6: On images of the real-val set, we visualize *ShoeRinsics*'s predictions including depth thresholding which generates predicted prints. Our method *ShoeRinsics* produces visually appealing intrinsic decompositions (depth, albedo, and normal). Importantly, on novel shoe-tread displayed in the last bottom rows, *ShoeRinsics* produces very good depth and shoeprints by comparing against the ground-truth shoeprints. To display the predicted prints, we threshold the predicted depth to best match the ground-truth print (Sec 2.3.2).

Mitigating domain gap by feature alignment. We further adopt the feature alignment technique to mitigate the domain gap [113]. Specifically, we learn an adversarial discriminator $\mathcal{D}_{feat}$ to discriminate *features* extracted by the decomposer for the real images and the translated synthetic images. We use this as a loss in training the decomposer and update the discriminator $\mathcal{D}_{feat}$ while training of the decomposer. This encourages the decomposer to extract features on real data that are indistinguishable from synthetic data, thus helping mitigate the domain gap.

Exploiting Pseudo Albedo. Shoe-treads, like many other man-made objects such as cars and other toys, tend to have piece-wise constant albedo. Building on this observation, we create pseudo albedo for the real data by grouping pixels with the mean-shift algorithm [31]. Fig. 2.4 shows an example pseudo albedo on two real shoes. As pseudo albedo is not ideal as ground-truth, we use it to learn an albedo predictor through the the decomposer. We find this produces better albedo maps than the pseudo ground-truth (see analysis in the supplement). To learn albedo prediction, we minimize the $\mathcal{L}_1$ loss, i.e., $\mathcal{L}_1(\hat{X}^a_R, \text{MS}(I_R))$, where MS is the mean-shift clustering algorithm.

Stage-wise Training is common in training multiple modules, particularly with GAN discriminators. Our training paradigm contains four stages. First, we train the decomposer $\mathcal{F}$ and renderer $\mathcal{R}$ on syn-train. Second, we train the image translators and discriminators $G_{S\to R}$, $\mathcal{G}_{R\to S}$, $\mathcal{D}_R$, and $\mathcal{D}_S$ with Eq. 2.2. Third, we finetune $\mathcal{R}$ using the translated synthetic images by $\mathcal{G}_{S\to R}$. Finally, we freeze $\mathcal{R}$ and $\mathcal{G}_{S\to R}$ and finetune $\mathcal{F}$ on translated synthetic images and real images using losses described above.

2.6 Experiments

We validate our *ShoeRinsics* and compare it against prior methods of depth prediction on our benchmark. We start with implementation details, followed by a visual comparison and quantita-

tive evaluation, and conduct an ablation study and analysis of why *ShoeRinsics* outperforms the prior art.

2.6.1 Implementation

Training specifics. Instead of using high-resolution images (405x765) from the training set, we crop patches (128x128) to train the models. We find this yields better performance, as shown in the ablation study (Sec. 2.6.4). For a fair comparison, we train all models with patches for the same number of optimization steps. During training, we sample patches from random positions. We use Adam optimizer and set the learning rate as 1e-3 and 1e-4 for training the initial models (e.g., $\mathcal{F}$ and $\mathcal{R}$) and finetuning them, respectively. We set the batch size as 8 throughout our experiments. Recall that we train our model in stages (Sec. 3.5). We train for 20M iterations in the first two stages and 100K iterations in the last two stages.

Architectures. Our decomposer $\mathcal{F}$ and renderer $\mathcal{R}$ have a classic encoder-decoder structure as used in [44]. We modify the light prediction decoder to be a 17-way classifier (given that our synthetic data has only 17 lighting configurations). We also add residual connections between layers to predict full-resolution maps for intrinsic components (depth, albedo, and normal). Our translators and discriminators ($\mathcal{G}_{S\to R}, \mathcal{G}_{R\to S}, \mathcal{D}_R, \mathcal{D}_S$, and $\mathcal{D}_{feat}$) have the same structure as used in [41]. The $\mathcal{D}_{feat}$ is a convolutional network that uses a kernel size 3 to process the albedo, depth, and normal features. It further takes as input the features of the lighting prediction branch. That said, $\mathcal{D}_{feat}$ learns to discriminate features of all the intrinsic components.

Hyperparameter setting. We denote the combined hyperparameters as $\hat{\lambda} = (\lambda_a, \lambda_d, \lambda_n, \lambda_l)$ in Eq. 2.1. The decomposer $\mathcal{F}$ is trained with $\hat{\lambda} = (1, 1, 1, 0.1)$ in the first stage, and finetuned with $\hat{\lambda} = (1, 2, 1, 0.1)$ in the final stage. When finetuning, we set the weight to 3 for the reconstruction loss, 2 for the pseudo albedo loss, and 1 for the feature alignment. We set the hyperparameters via validation.

Test-time augmentation. During testing, we consider test-time augmentation [28, 40]. For each image, we produce 23 variants: 3 flips (horizontal, vertical, and vertical+horizontal), 4 rotations (angles $+5°$, $+10°$, $-5°$, and $-10°$), 4 scalings (scale factor 0.5, 0.8, 1.5, and 1.8), and 12 flip+rotation versions (three flips times four rotations). For each variant, we predict the depth and then transform back to the original coordinate frame. We average all the 24 depth maps as the final prediction.

2.6.2 Qualitative Results of ShoeRinsics

We visualize predictions on the real-val images by our method *ShoeRinsics* in Fig. 2.6. *ShoeRinsics* predicts good depth maps, the thresholding of which generates shoeprints that match the ground-truth prints. As a byproduct, our method also makes visually appealing predictions on other intrinsic components. We compare our predictions with those made by other methods on real-val (Fig. 2.7) and real-FID-val (Fig. 2.8). Clearly, our *ShoeRinsics* produces more reasonable visuals (depth and shoeprints) than the compared methods. The supplement has further visualizations.

2.6.3 State-of-the-art Comparison

ShoeRinsics outperforms prior methods in most of the validation examples (details in the supplement). Table 2.2 and 2.3 list comparisons as analyzed below.

Comparison with intrinsic image decomposition. We compare our *ShoeRinsics* and RIN [44], which learns for intrinsic image decomposition. As RIN [44] emphasizes normal prediction to represent shapes, we use the standard Frankot-Chellappa algorithm [29] to integrate the normals towards depth maps. Compared to [44], our *ShoeRinsics* explicitly incorporates domain adaptation in the image and feature space. Doing so helps mitigate the synthetic-real domain gap. As a result, *ShoeRinsics* outperforms RIN on both real-val and real-FID-val (Table 2.2 and 2.3). On real-val,

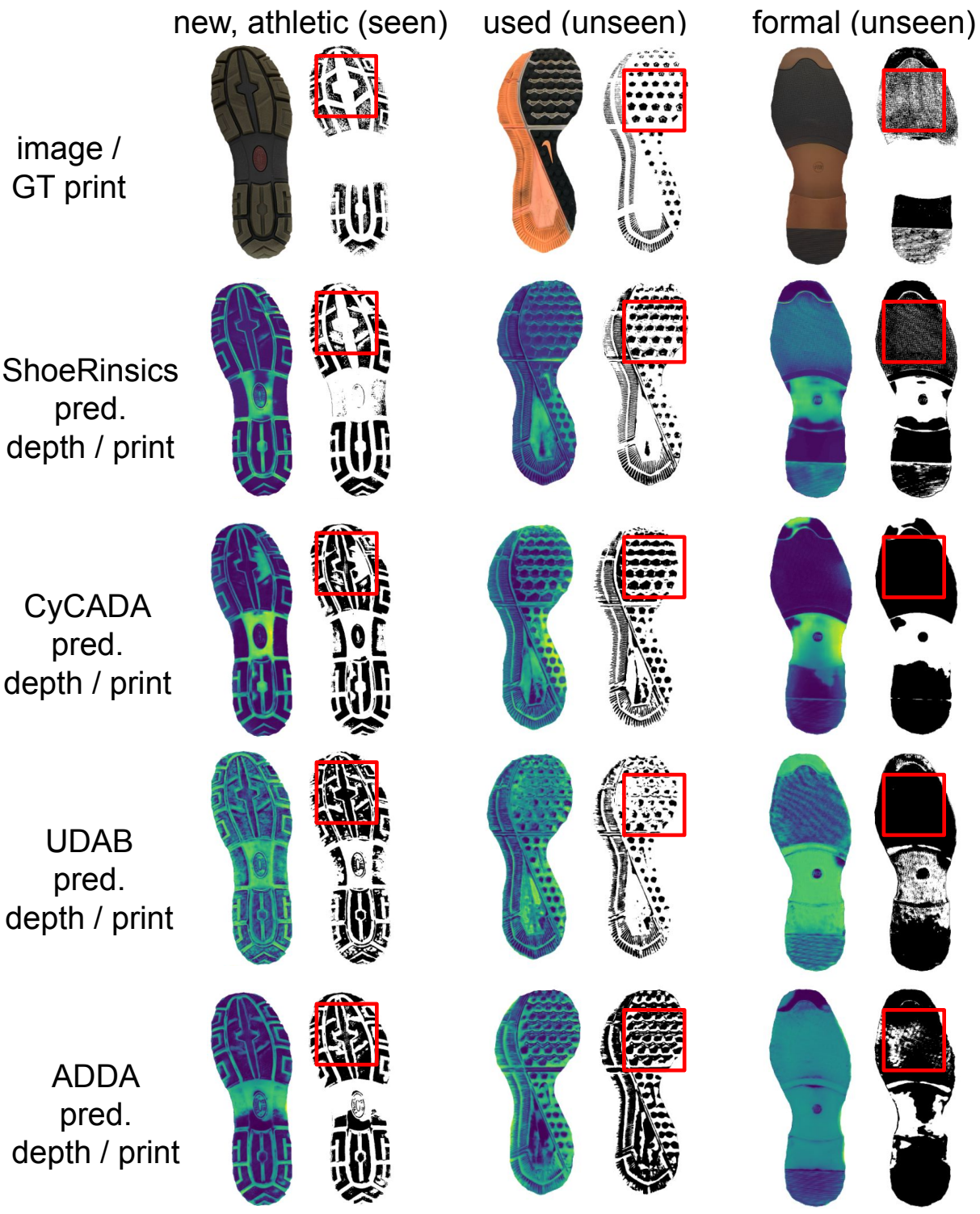

Figure 2.7: Comparison with the state-of-the-art methods of domain adaptation tailored to depth prediction on our real-val benchmark. Our *ShoeRinsics* performs better than others for both seen and unseen shoe categories as highlighted by the red boxes.

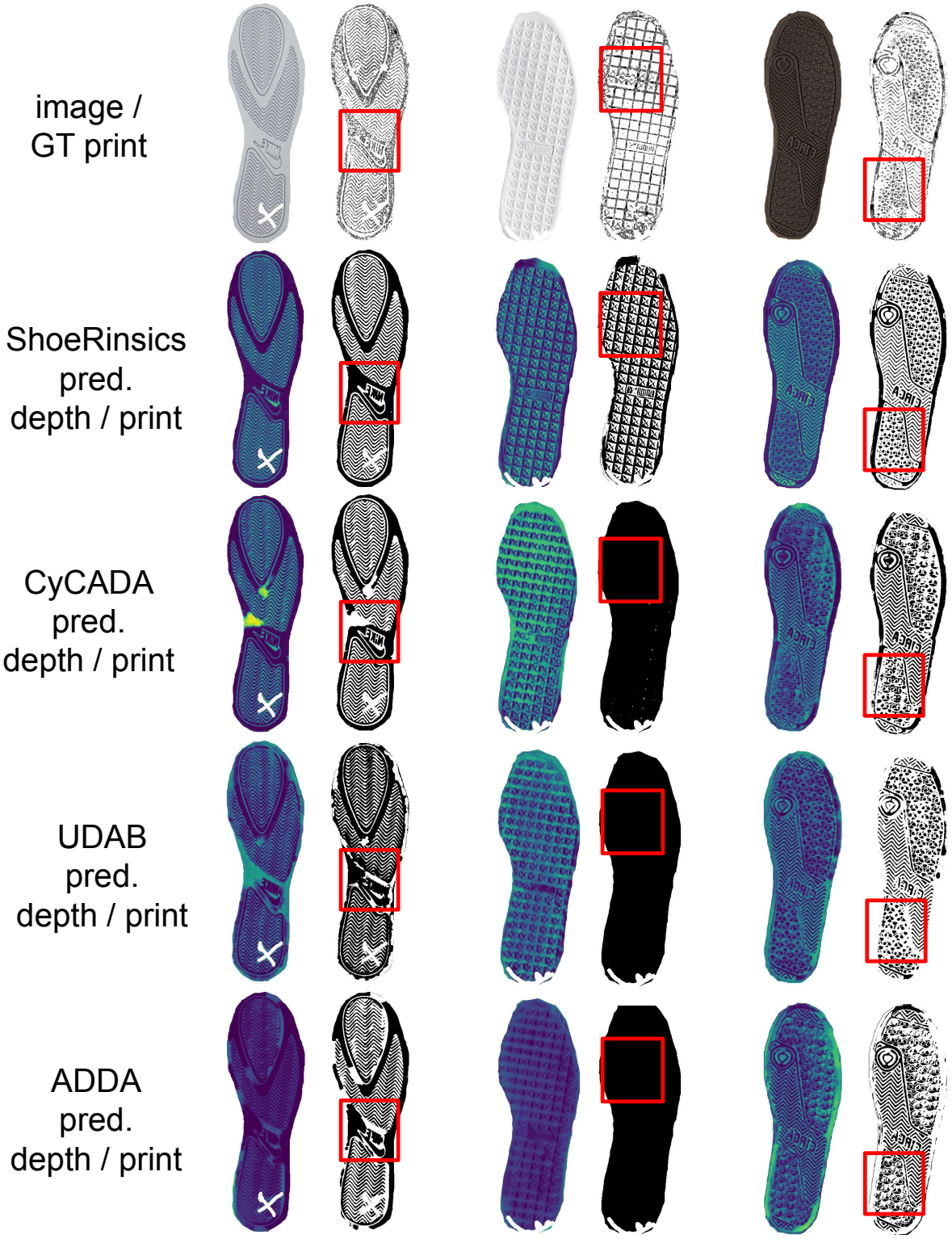

Figure 2.8: Comparison with the state-of-the-art methods for depth prediction and domain adaptation [33, 41, 93] on real-FID-val. Clearly, *ShoeRinsics* produces shoeprints which are visually closer to the ground-truth than previous methods.

Table 2.2: Benchmarking on real-val. We use IoU as the metric (in %), and break down the analysis for different shoe categories (*new-athletic* shoes seen during training, and *formal* and *used* shoes unseen in training). We compute mean IoU (mIoU) over all validation examples. Training on only synthetic data yields poor performance, whereas our *ShoeRinsics* performs the best on both seen and unseen categories. This clearly demonstrates the benefit of combining synthetic-to-real domain adaptation with intrinsic decomposition. The ablation study (bottom panel) shows that each individual component (discriminator, translator, and renderer, cf. Fig. 2.1) helps improve shoeprint prediction. Lastly, from our syn-only ablation, decomposing to all intrinsic components performs better than training a depth predictor for shoeprint prediction, further demonstrating that incorporating intrinsic decomposition helps close synthetic-to-real domain gaps. Exploiting test-time augmentation boosts performance from *mIoU* = 46.8 to 49.0.

Method	*New-Athletic (seen)*	*Formal (unseen)*	*Used* (unseen)	*mIoU*
RIN [44]	30.0	39.7	24.4	30.4
ADDA [93]	46.5	41.4	27.2	41.4
UDAB [33]	46.0	40.4	29.6	41.4
CyCADA [41]	48.8	43.9	34.5	44.8
syn-only, depth only	41.3	41.2	28.4	38.4
syn-only, all intrinsics	41.8	41.5	27.1	38.5
ShoeRinsics	50.5	47.8	35.8	46.8
w/o discriminator	48.2	39.9	33.6	43.6
w/o translator	49.0	42.8	31.4	44.0
w/o renderer	49.0	46.4	34.7	45.4
ShoeRinsics w/ aug	**52.4**	**52.9**	**36.9**	**49.0**

it performs better than RIN by 20.5% mIoU on the (*seen new-athletic*) shoes, by 8.1% mIoU on the *formal unseen* shoes, by 11.4% mIoU on *used unseen* shoes. On real-FID-val, *ShoeRinsics* improves IoU by 5.6% mIoU over RIN.

Comparison with domain adaptation. Table 2.2 and 2.3 clearly show that our *ShoeRinsics* consistently outperforms the compared domain adaptation methods (ADDA [93], UDAB [33], and CyCADA [41]) on both the real-val and real-FID-val datasets. From ablation studies, as shown in the lower panel of Table 2.2, we see that using the renderer (cf. Fig. 2.1) and the decomposer (that learns to predict albedo, normal, and lighting as auxiliary supervisions) greatly improves the performance. Qualitative comparison on real-val in Fig. 2.7 and real-FID-val in Fig. 2.8 show that depth maps and the corresponding prints predicted by our *ShoeRinsics* have richer textures and

Table 2.3: Benchmarking on real-FID-val. We report mean IoU (mIoU) over validation examples. *ShoeRinsics* outperforms previous methods and improves further with test-time augmentation.

	RIN [44]	ADDA [93]	UDAB [33]	CyCADA [41]	**ShoeRinsics**	**ShoeRinsics** w/ aug
mIoU	26.0	27.2	29.0	31.2	31.6	**32.0**

Figure 2.9: Comparison between real-FID-val (a) and real-val (b). The shoeprints from real-FID-val are noisy and slightly misaligned with the corresponding shoe-treads. In contrast, shoeprints of real-val contain the entire contact surfaces and are well aligned with the corresponding shoe-tread images.

better-aligned patterns to the RGB input. When exploiting test-time augmentation (cf. *ShoeRinsics* w/ test-time aug), we boost the performance from mIoU = 46.8% to 49.0% on real-val and from mIoU=31.6% to 32.0% on real-FID-val.

Performance on real-val vs real-FID-val. All the methods show lower mIoU numbers on real-FID-val compared to real-val. This is owing to the noisy ground-truth prints of real-FID-val (see Fig. 2.9). Note that the FID prints are obtained by pressing gelatin lifters onto dusty shoe-treads followed by scanning the lifters [50]. This means that the shoeprints can be noisy as the contact surfaces do not leave a full print. In contrast, for real-val shoeprints, we minimize such noise and get more even coverage by averaging over multiple prints for the same shoe. Moreover, while real-val consists of image and print pairs of the exact same shoe, real-FID-val consists of prints from [50] with our manually discovered shoe-tread images, meaning that they might not be well aligned, as visually seen in Fig. 2.9.

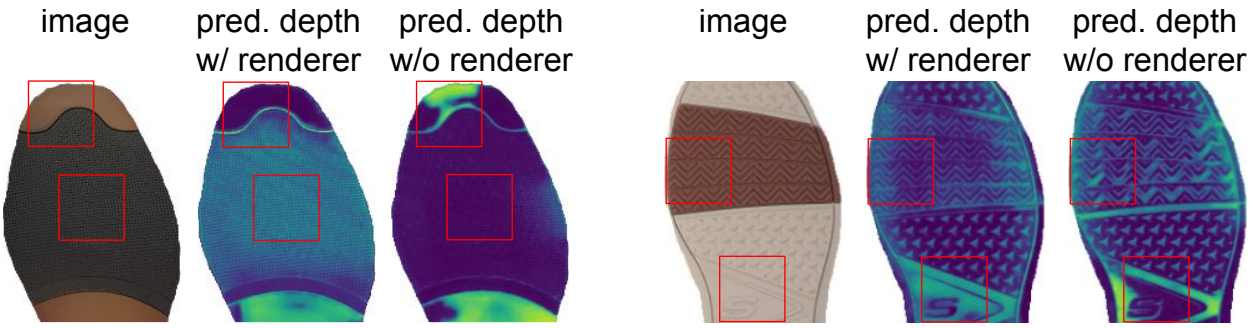

Figure 2.10: Training *ShoeRinsics* with the renderer (which allows using the reconstruction loss) produces visibly better depth than without. Using the renderer encourages the decomposer to output depth maps that contain fine-grained details because it penalizes coarse predictions through the image reconstruction loss. That said, the renderer regularizes the learning for depth prediction by exploiting auxiliary supervisions from other intrinsic components (albedo, normal, and lighting).

2.6.4 Ablation Study

We conduct an ablation study (cf. Table 2.2 bottom panel) on the modules in *ShoeRinsics*, including feature alignment (by learning discriminator $\mathcal{D}_{feat}$ in the feature space), translator $\mathcal{G}_{S\to R}$, and renderer $\mathcal{R}$. All three modules aim to mitigate synthetic-real domain gaps. We also study whether predicting intrinsic components (albedo, normal, and lighting) helps depth prediction and whether patch-based learning is better than full-image learning.

Effect of feature alignment by discriminator $\mathcal{D}_{feat}$. *ShoeRinsics w/o discriminator* removes the feature discriminator $\mathcal{D}_{feat}$ but keeps all the other modules. It yields 43.6% mIoU, 3.2% mIoU lower than *ShoeRinsics* (cf. Table 2.2). This demonstrates the effectiveness of $\mathcal{D}_{feat}$ for mitigating domain gaps by aligning features.

Effect of image translator $\mathcal{G}_{S\to R}$. *ShoeRinsics w/o translator* drops the translators but keeps other components, achieving 44.0% mIoU, 2.8% mIoU lower than *ShoeRinsics* (cf. Table 2.2). This shows the effectiveness of using translators to close the synthetic-real domain gap.

Effect of the reconstruction loss by the renderer $\mathcal{R}$. *ShoeRinsics w/o renderer* drops the renderer from *ShoeRinsics*, leading to 45.4% mIoU, 1.4% mIoU lower than *ShoeRinsics* (cf. Table 2.2).

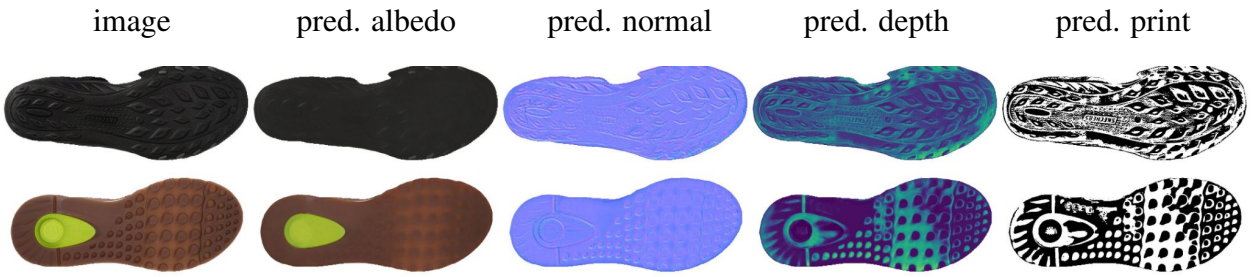

Figure 2.11: Failure cases. *ShoeRinsics* performs poorly in the presence of complex materials (e.g., translucence).

This validates the effectiveness of the renderer. Fig. 2.10 visualizes depth predictions with and without the renderer during training. Clearly, with the renderer, the predicted depth has better high-frequency textures. See the caption of Fig. 2.10 for details.

All intrinsics vs depth only. Comparing "syn-only, depth only" and "syn-only, all intrinsics" in Table 2.2, we see that learning to predict all intrinsics performs slightly better (38.5% vs. 38.4%). Importantly, this allows using the renderer as the reconstruction loss to regularize the training on real images, yielding significantly better results in the final *ShoeRinsics* (46.8% mIoU).

Patches vs. full-resolution images. We compare the depth prediction performance by training the decomposer on patches versus full-resolution images of the synthetic data. We find that the former (patch-based) achieves 38.5% mIoU (cf. Table 2.2) as opposed to 36.5% mIoU for the latter (not shown in the table). This demonstrates the benefit of depth learning on patches over whole images in this setup.

2.6.5 Failure Cases

We analyze failure cases of *ShoeRinsics* in Fig. 2.11. We find that our method performs poorly on shoes with complex materials. One reason is that the syn-train data does not contain any complex materials. Future work may explore richer synthetic datasets to improve performance.

2.7 Conclusion

Motivated by constructing a database of shoeprints for forensic use, we introduce a problem of predicting depth for shoe-tread photos collected by online retailers. Because these photos do not have ground-truth depth, we exploit synthetic images (containing shoe-treads and ground-truth intrinsics including depth, albedo, normal, and lighting). We study domain adaptation and intrinsic image decomposition techniques and propose a method termed *ShoeRinsics* to train for depth prediction. Our experiments demonstrate consistent improvements of *ShoeRinsics* over previous methods on this task. We expect future algorithmic explorations on this task from the perspective of domain adaptation, depth prediction, and intrinsic decomposition.

Chapter 3

CriSp: Leveraging Tread Depth Maps for Enhanced Crime-Scene Shoeprint Matching

3.1 Introduction

Examining the evidence found at a crime scene assists investigators in identifying suspects. Shoeprints are more likely to be found at crime scenes, though they may possess fewer distinct identifying features compared to other biometric samples like blood or hair [16]. Consequently, analyzing shoeprints can furnish crucial leads to aid investigators in narrowing down potential suspects in a crime.

The examination of shoeprints forensically offers insights into both the class attributes and the acquired attributes of the suspect's footwear. Class attributes pertain to the general features of the shoe, such as its brand, model, and size. On the other hand, acquired attributes encompass the unique traits that develop on a shoe with wear and tear, such as holes, cuts, and scratches. Our focus lies in facilitating the investigation of the class attributes of shoeprints.

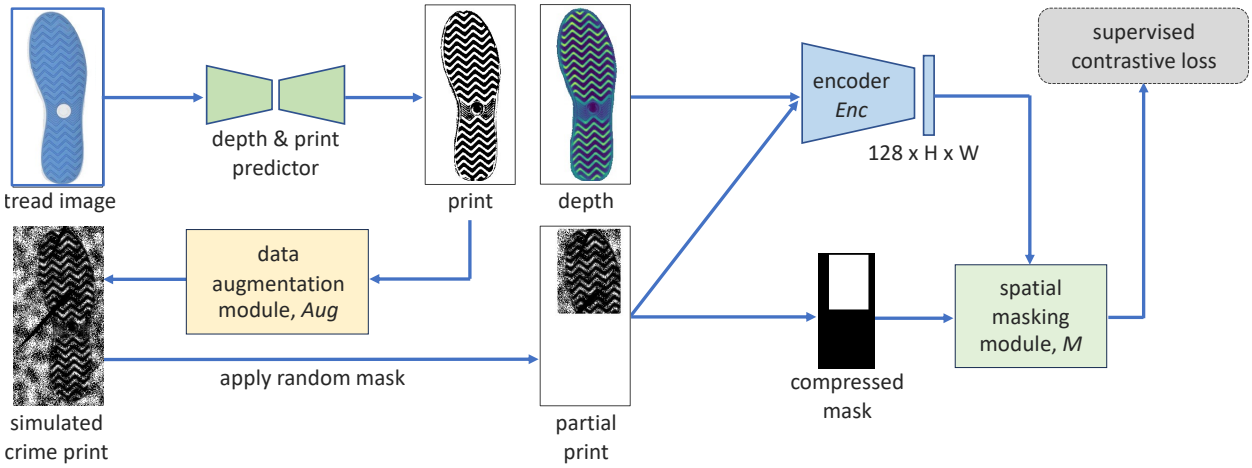

Figure 3.1: We develop a method termed *CriSp* to compare crime-scene shoeprints against a database of tread depth maps (predicted from tread images collected by online retailers) and retrieve a ranked list of matches. We train *CriSp* using tread depth maps and clean prints (Sec. 3.4) as shown above. We use a data augmentation module Aug to address the domain gap between clean and crime-scene prints, and a spatial feature masking strategy (via spatial encoder Enc and masking module M) to match shoeprint patterns to corresponding locations on tread depth maps (Sec. 3.5). *CriSp* achieves significantly better retrieval results than the prior methods (Sec. 3.6)

Status quo. Traditional automated shoeprint matching methods [17, 5, 97, 49, 22, 36, 51, 7, 8] typically use handcrafted priors to match crime-scene shoeprints with clean, reference impressions. Recent approaches [107, 48, 63] propose using more generalizable features from Convolutional Neural Networks (CNNs), yet they are limited to using CNNs pretrained on ImageNet [23] since available datasets [50] are small and not suitable for training. Training on shoeprint-specific data is expected to improve matching performance. Importantly, while existing methods match crime-scene shoeprints to clean reference shoeprints, our findings suggest that matching to tread depth maps containing 3D shape information is more effective (see details in Sec. 3.6.3).

Motivation. To address the need for a large-scale training dataset, our proposal leverages the extensive collection of tread images of various shoe products available from online retailers. Tread depth maps and clearly visible prints can be predicted from these images using [81]. Example tread images and their depth and print predictions are illustrated in Fig. 3.3. It is important to note that matching directly to RGB tread images causes models to overfit to irrelevant details such as albedo and lighting (see Sec. 3.6.3). *We formulate our problem as the retrieval of tread depth maps that*

best match crime-scene shoerpints by learning a representation from tread depth maps and clean shoeprints.

Technical insights. We develop a method termed *CriSp* to address this problem using 3 key components (Fig. 3.1). Firstly, we employ a data augmentation module Aug to generate simulated crime-scene shoeprints from the clearly visible prints used during training. In an ideal scenario, learning a feature representation to match crime-scene shoeprint images to tread depth maps would require a dataset containing paired crime-scene shoeprints and corresponding tread depth maps. In the absence of such data, our data augmentation module in combination with our training set of paired tread depth maps and clean prints serves as a viable alternative. Secondly, we propose a spatial encoder Enc to ensure that our model learns to match patterns in corresponding regions of shoe treads. For instance, if a crime-scene shoeprint exhibits stripes on the heels, the model must retrieve shoes with stripes in the heel region rather than other areas like the toe region. Thirdly, we incorporate a feature masking module M to ensure that only the visible portions of shoeprints affect our retrieval results when crime-scene shoeprints are only partially visible. We observe that combining these three techniques facilitates effective feature learning and results in significantly improved retrieval performance compared to existing state-of-the-art methods.

Contributions. This chapter makes three major contributions.

- We introduce the concept of matching crime-scene shoeprints to tread depth maps, leading to improved retrieval performance compared to traditional shoeprint matching methods.
- We propose a benchmarking protocol to evaluate our method against state-of-the-art approaches. This involves the creation of a new dataset, reprocessing of existing datasets, and defining appropriate evaluation metrics.
- We develop a spatially-aware matching method named *CriSp*, which demonstrates superior performance over existing methods in both shoeprint matching and image retrieval tailored to this specific task.

3.2 Related Work

Automated shoeprint matching. The success of automated fingerprint identification systems [21] has inspired researchers to study automated shoeprint matching. Current literature aims to extract features from crime-scene shoeprints and match them to a database of laboratory footwear impressions to identify the shoe make and model [74]. Holistic methods process the shoeprint image as a whole. Examples of this method use Hu's moment [5], Zernike moment [97], and Gabor and Zernike features [49]. In contrast, local methods extract discriminative features from local regions of the shoeprint, making them more adept at handling partial prints. For instance, [51] exploit Wavelet-Fourier transform features, [7] introduce a block sparse representation technique, and [8] combine the Harris and the Hessian point of interest detectors with SIFT descriptors. Recent works [107, 48, 63] propose using features from networks pretrained on ImageNet [23] for their generalizability. However, the lack of large-scale shoeprint datasets hampers their effectiveness. To address this, we propose creating a large-scale training dataset by leveraging tread images from online retailers and utilizing an off-the-shelf predictor [81] to estimate their depth and print. This approach enables learning features from informative tread depth maps rather than shoeprints, ensuring good performance even with partially visible prints.

Image retrieval. Image retrieval techniques have been a popular research problem for several decades [111]. Traditional methods use handcrafted local features [62, 14], often coupled with approximate nearest-neighbor search methods using KD trees or vocabulary trees [15, 39, 65, 68]. More recently, the success of CNNs in classification tasks [54] encouraged their use in image retrieval tasks [12, 83]. Global features can be generated by aggregating CNN features [11, 91, 10, 35, 70, 90, 66, 71, 98], while local features can also be used for spatial verification [68, 66, 19, 98, 47] which ensure better performance by using geometric information of objects. Our problem differs from this category of work since our query and database data come from different domains - crime-scene shoeprints and depth maps of shoe treads. Even within our query set of crime-scene shoeprints, images can be from various sources such as blood, dust, and sand impressions.

Cross-domain image retrieval. More closely related to our work is cross-domain image retrieval (CDIR), where the query and database images come from different domains. The fundamental idea is to map both domains into a shared semantic feature space to alleviate the cross-domain gap. Learning a distinct representation for each shoe model can be categorized as fine-grained cross-domain image retrieval (FG-CDIR) as we aim to retrieve one instance from a gallery of same-category images. It is harder than category-level classification [24, 26, 103] tasks since the differences between shoe treads are often subtle. A popular problem of this category, fine-grained sketch-based image retrieval (FG-SBIR), was introduced as a deep triplet-ranking based siamese network [69] for learning a joint sketch-photo manifold. FG-SBIR was improvised via attention-based modules with a higher order retrieval loss [87], textual tags [20, 86], and hybrid cross-domain generation [67]. Recently, [76] leveraged a foundation model (CLIP) and [60] explicitly learned local visual correspondence between sketch and photo to offer explainability. These works differ from ours in that we do not have any ground-truth training data from our query domain, and thus have to simulate it as best as we can. Additionally, our aligned query and database images enable us to use spatially-aware techniques like spatial feature masking.

3.3 Problem Setup and Evaluation Protocol

Our goal is to retrieve shoe models that best match crime-scene impressions by comparing against a comprehensive shoe collection. We propose using tread images from online retailers to build our reference database. The problem formulation and evaluation protocol is outlined below.

3.3.1 Problem setup

Tread depth maps are more informative and relevant than RGB tread images (refer to Sec. 3.6.3). Since there is no dataset of crime-scene shoeprints with ground-truth tread depth maps, we propose

to learn from tread depth maps and clean shoeprints predicted from RGB tread images (details in Sec. 3.4.1) instead. A crime-scene investigator may want to use our method to get a ranked list of shoe models that match a crime-scene shoeprint and opt to manually inspect them. Therefore, **the problem is to generate a ranking $[r_1, r_2, ..., r_n]$ of shoe models from a reference database of tread depth maps with shoe model tags where r_i is more likely to be the shoe model that left a crime-scene shoeprint than r_j for all $i < j$ by learning a representation from a dataset of tread depth maps and clean, fully-visible prints**. This requires (1) addressing the domain gap between crime-scene shoeprints and clean prints, and (2) matching patterns to corresponding locations of shoe treads.

3.3.2 Evaluation Protocol

To benchmark methods, we introduce two validation sets of crime-scene shoeprints with ground-truth shoe model labels, which are linked to a large-scale reference database (see details in Sec. 3.4.2). Note that the ground-truth for a shoeprint may contain multiple shoe models since tread patterns can be shared by different shoe models. In practice, we expect crime-scene investigators to look through the top K retrieved shoe models and we set K to be a realistically small value of 100, representing the top 0.4% shoe models in our reference database. We use two metrics to compare models based on their top K retrievals. Our first metric, mean average precision at K (mAP@K), is a standard metric to compare ranking performance. It considers both the number of positive matches and their positions in the ranking list. The second metric, hit ratio at K (hit@K), is more intuitive and represents the fraction of times we get at least one positive match in the top K retrievals. This metric is useful because a positive match can be used in a query expansion step to retrieve other good matches much more effectively [27]. Both metrics have values between 0 and 1, with higher numbers representing better performance. The supplement has further details.

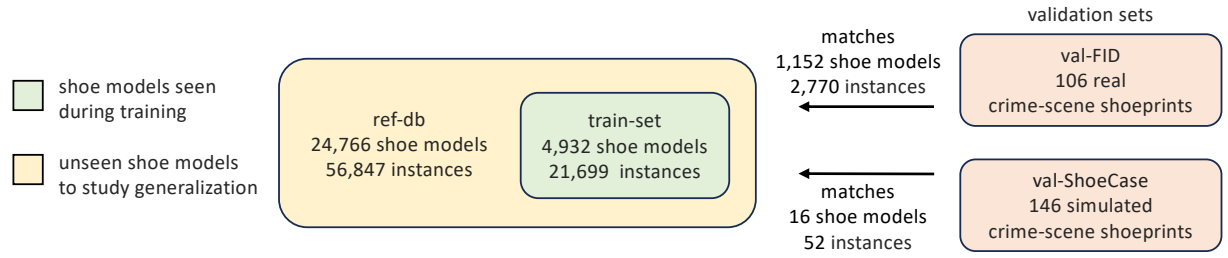

Figure 3.2: Dataset statistics. We have a reference database (ref-db) and two validation sets (val-FID and val-ShoeCase) with crime-scene impressions to query against ref-db. We use a section of ref-db for training (train-set) and leave the rest to study generalization. Ground-truth labels from our validation sets connect our query crime-scene shoeprints to shoes in ref-db. See details in Sec. 3.4 and visual examples in Fig. 3.3 and 3.4.

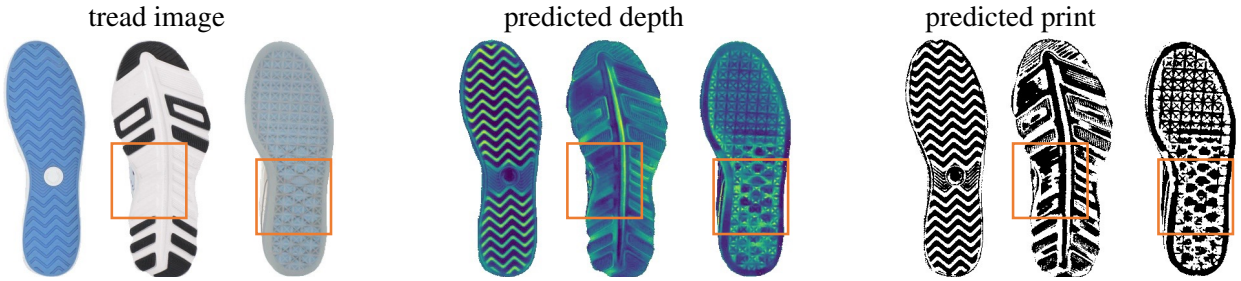

Figure 3.3: Examples from train-set. We create training data by leveraging shoe-tread images available from online retailers and predicting their depth maps and prints as outlined in [81]. Note that the depth and print predictions are not always accurate (2nd and 3rd shoe).

3.4 Dataset Preparation

We train our model on a dataset (train-set) of aligned shoe tread depth maps and clean shoeprints. To study the effectiveness of models, we introduce a large-scale reference database (ref-db) of tread depth maps, along with two validation sets (val-FID and val-ShoeCase) created by reprocessing existing datasets of crime-scene shoeprints [50, 89]. We match shoeprints from the validation sets to ref-db and add labels connecting shoeprints in val-FID and val-ShoeCase to ref-db to enable quantitative analysis. An overview of the datasets is provided in Fig. 3.2, while Fig. 3.3 and Fig. 3.4 present example depth maps, clean prints, and crime-scene prints. In this section, we elaborate on our training dataset (train-set), reference database (ref-db), and validation sets (val-FID and val-ShoeCase).

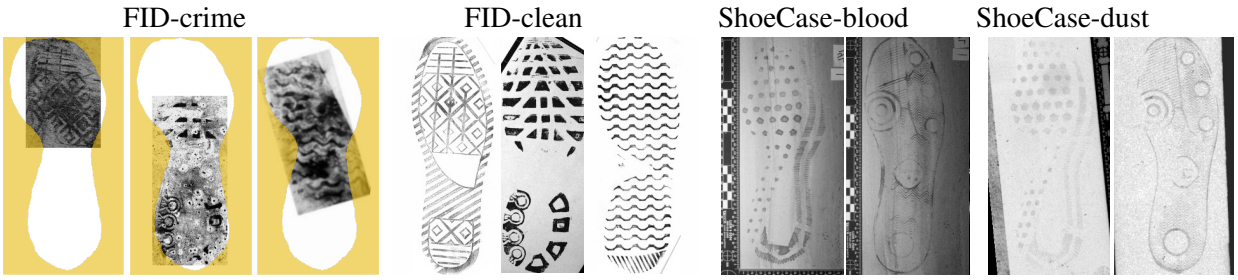

Figure 3.4: Examples from val-FID and val-ShoeCase. Val-FID contains real crime-scene prints (FID-crime) and clean, fully visible lab impressions (FID-clean). We show FID-crime and FID-clean shoeprints corresponding to the same shoe models for easier comparison. Note that we show a yellow shoe outline on the FID-crime prints for visualization purposes and the outline does not exist in FID-crime images. Val-ShoeCase contains simulated crime-scene shoeprints on blood (ShoeCase-blood) and dust (ShoeCase-dust). All val-ShoeCase prints are full-sized, as opposed to val-FID.

3.4.1 Online Shoe Tread Depth Maps and Prints for Training

Train-set. Online retailers [1, 3] showcase images of shoe treads for advertisement. Our training set (train-set) contains depth maps and clean, fully visible prints from such tread images as predicted by [81]. We also apply segmentation masks as determined by [81] to the predictions. To ensure consistency across all images, we employ a global alignment method to minimize variations in scale, orientation, and center using a simple model. Sample shoe-tread images along with their corresponding depth and print predictions are illustrated in Fig. 3.3. Online retailers categorize shoe styles using stock keeping units (SKUs), which we use as shoe model labels. Shoes with the same SKUs can have different colors and sizes. Sometimes different shoe models can share the same tread pattern.

Statistics. Train-set contains 21,699 shoe instances from 4,932 different shoe models. Each shoe model in our database can have shoe-tread images from multiple shoe instances, possibly with variations in size, color, and lighting. The tread images in train-set have a resolution of 384x192.

Inaccuracies. It is important to note that the training dataset can have some inaccuracies since it comes from raw data downloaded from online retailers. Some tread images might have incorrect model labels, and some images may not depict shoe treads. Other inaccuracies come from imper-

fect depth and print prediction (cf. Fig. 3.3), segmentation errors, and alignment failures. We hope to mitigate the errors by including multiple instances per shoe model in train-set.

3.4.2 Reference Database and Crime-scene Shoeprints for Validation

Ref-db. We introduce a reference database (ref-db) by extending train-set to include more shoe models. The added shoe models are used to study generalization to unseen shoe models. Ref-db contains a total of 56,847 shoe instances from 24,766 different shoe models. The inclusion of multiple instances per shoe model in ref-db allows the depth predictor some margin for error (cf. Fig. 3.3), ensuring minimal impact on the overall matching algorithm performance since it has multiple chances to match a query print to a shoe model. The supplement has details on the distribution of shoe models from our validation sets in ref-db.

Val-FID. We reprocess the widely used FID300 [50] to create our primary validation set (val-FID). Val-FID contains real crime-scene shoeprints (FID-crime) and a corresponding set of clean, fully visible lab impressions (FID-clean). Examples of these prints are shown in Fig. 3.4. The FID-crime prints are noisy and often only partially visible. It contains impressions made by blood, dust, etc on various kinds of surfaces including hard floors and soft sand. To ensure alignment with ref-db, we preprocess FID-crime prints by placing the partial prints in the appropriate position on a shoe "outline" (cf. Fig. 3.4), a common practice in shoeprint matching during crime investigations.

We manually found matches to 41 FID-clean prints in ref-db by visual inspection. These are all unique tread patterns and correspond to 106 FID-crime prints. Given that multiple shoe models in ref-db can share the same tread pattern, we store a list of target labels for each shoeprint in FID-crime. These labels correspond to 1,152 shoe models and 2,770 shoe instances in ref-db (cf. Fig. 3.2).

Val-ShoeCase. We introduce a second validation set (val-ShoeCase) by reprocessing ShoeCase [89] which consists of simulated crime-scene shoeprints made by blood (ShoeCase-blood) or dust (ShoeCase-dust) as shown in Fig. 3.4. These impressions are created by stepping on blood spatter or graphite powder and then walking on the floor. The prints in this dataset are full-sized, and we manually align them to match ref-db.

ShoeCase uses two shoe models (Adidas Seeley and Nike Zoom Winflow 4), both of which are included in ref-db. The ground-truth labels we prepare for val-ShoeCase include all shoe models in ref-db with visually similar tread patterns as these two shoe models since we do not penalize models for retrieving shoes with matching tread patterns but different shoe models. Val-ShoeCase labels correspond to 16 shoe models and 52 shoe instances in ref-db (cf. Fig. 3.2).

3.5 Methodology

In this section, we introduce *CriSp*, our representation learning framework to match crime-scene shoeprint images S to tread depth maps d. An overview of our training pipeline is shown in Fig. 3.1. *CriSp* is trained using a dataset of globally aligned tread depth maps d and clean, fully-visible shoeprints s (see details in Sec. 3.4.1). The main components of our pipeline are a data augmentation module Aug to simulate crime-scene shoeprints, an encoder network Enc to map depths and prints to a spatial feature representation, and a spatial masking module M to mask out irrelevant portions from partially visible shoeprints.

Data augmentation. Our data augmentation module Aug simulates noisy and occluded crime-scene shoeprints (cf. Fig. 3.4) from clean, fully-visible prints (cf. Fig. 3.3), denoted as $\hat{S} = Aug(s)$. Aug uses 3 kinds of degradations (occlusion, erasure, and noise) as visualized in Fig. 3.5. Occlusion can be in the form of overlapping prints or random shapes. Erasures achieve the grainy

texture of crime-scene prints and noise adds background clutter to the images. Further details are provided in the supplement.

Encoder for spatial features. Our encoder Enc maps tread depths d and simulated crime-scene shoeprints $\hat{S}$ to a feature representation z, denoted as $z = Enc(x)$ where $x \in [d, \hat{S}]$. Enc consists of a modified ResNet50 [40] with the final pooling and flattening operation removed followed by a couple of convolution layers. Enc produces features of shape $[C, H, W]$ where C is the feature length, and H and W are the encoded height and width, respectively. For *CriSp* we set $C = 128$. Since our training data and query prints are globally aligned (cf. Sec. 3.4), Enc allows access to features at each (course) spatial location of the image, facilitating comparisons in corresponding locations of shoe treads. Enc has two input channels for depth and print, respectively. It processes only one input at a time and pads the other input channel with zeros.

Spatial feature masking. During training, we simulate partially visible crime-scene shoeprints by applying a random rectangular mask m to query prints. Our feature masking module M applies a corresponding mask to spatial features z to obtain $\bar{z} = M(z, m)$. M resizes mask m to a dimension of $[H, W]$, uses it to zero out spatial features outside the mask, and normalizes the masked features. This allows our model to focus on the visible portion of the prints. While it would make sense to apply mask m to tread depth images as well, we opt not to do this as it would necessitate recomputing all the database depth features for each query print image at inference time, which is not scalable.

Training loss and similarity metric. We train our model using supervised contrastive learning [47], which extends self-supervised contrastive learning to a fully supervised setting to learn from data using labels. For a set of N depth/print pairs $\{d_k, s_k\}_{k=1...N}$ from shoe models $\{l_k\}_{k=1...N}$ within a batch, and a randomly generated mask m per batch, we compute masked spatial features $\{\bar{z}_i\}_{i=1...2N}$ and corresponding shoe labels $\{\bar{l}_i\}_{i=1...2N}$ where $\bar{z}_{2k} = M(Enc(d_k), m)$, $\bar{z}_{2k+1} = M(Enc(Aug(s_k)), m)$, and $\bar{l}_{2k} = \bar{l}_{2k+1} = l_k$. We treat $\bar{z}$ as a vector of size CHW and apply the

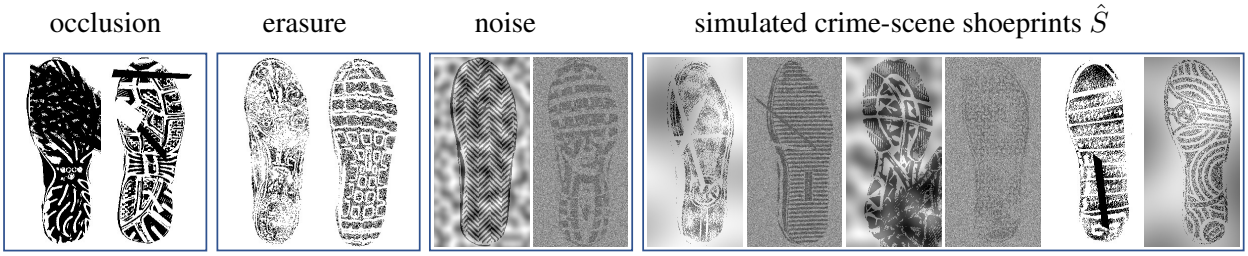

Figure 3.5: Examples of data augmentation. Our data augmentation module Aug simulates crime-scene shoeprints (cf. Fig. 3.4) from clean, fully visible prints in our training set (cf. Fig. 3.3). Aug optionally (1) introduces occlusion such as overlapping prints and random shapes, (2) erases parts of the print to create a grainy appearance, and (3) adds noise to mimic background clutter.

following loss.

$$\mathcal{L} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\bar{z}_i \cdot \bar{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\bar{z}_i \cdot \bar{z}_a / \tau)} \tag{3.1}$$

Here, $i \in I \equiv \{1 \dots 2N\}$, $A(i) \equiv I \setminus \{i\}$, and $P(i) \equiv \{p \in A(i) : \bar{l}_p = \bar{l}_i\}$ is the set of indices of all positives in the batch distinct from i. $|P(i)|$ is the cardinality of $P(i)$. The $\cdot$ symbol denotes the inner product, and $\tau \in \mathcal{R}^+$ is a scalar temperature parameter. This loss corresponds to using cosine similarity to measure similarity between images.

Sampling. For the above loss to be effective, we must have positive examples for all shoe models within a batch. However, if we pick shoe models randomly from a dataset with a very large number of shoe models, we can expect to sample a set of unique shoe models each time. Then, this loss would act like its self-supervised counterpart. To remedy this, we sample data in pairs, i.e. we choose $N/2$ shoe models randomly and select two shoe instances from each shoe model.

3.6 Experiments

We evaluate our *CriSp* and compare it with state-of-the-art methods on automated shoeprint matching [48] and image retrieval [47, 98, 76, 60]. We begin with visual comparison and quantitative

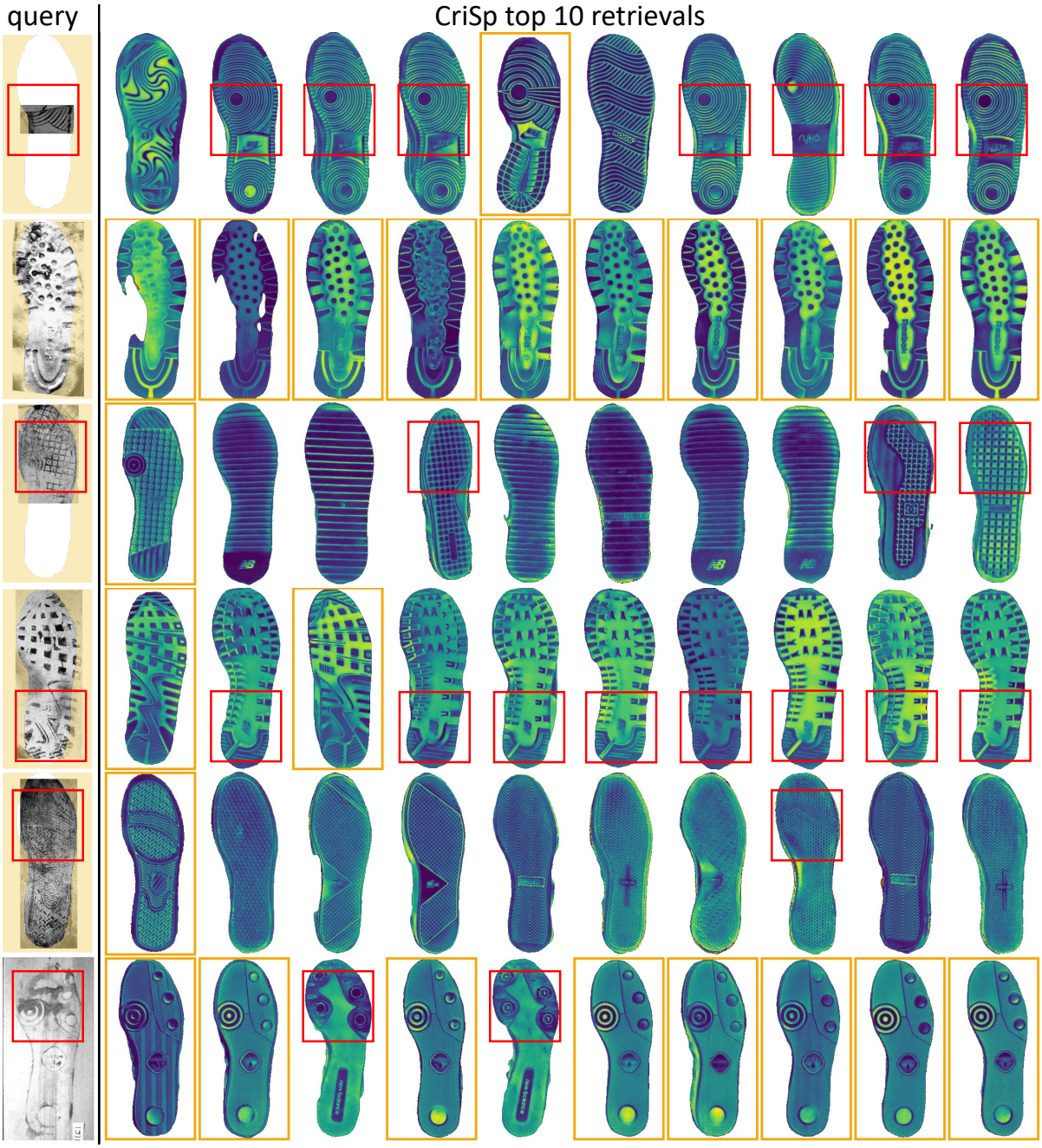

Figure 3.6: Visualization of the top 10 retrievals by *CriSp* on val-FID (rows 1-5) and val-ShoeCase (row 6). *CriSp* retrieves positive matches (highlighted in orange) very early even when crime-scene shoeprints have very limited visibility or severe degradation. Additionally, corresponding locations on the retrieved shoes share similar patterns to the query print, even in negative matches (highlighted in red).

evaluation, followed by an ablation study and analysis of our design choices. We will open-source our code and dataset to foster research after acceptance of this work.

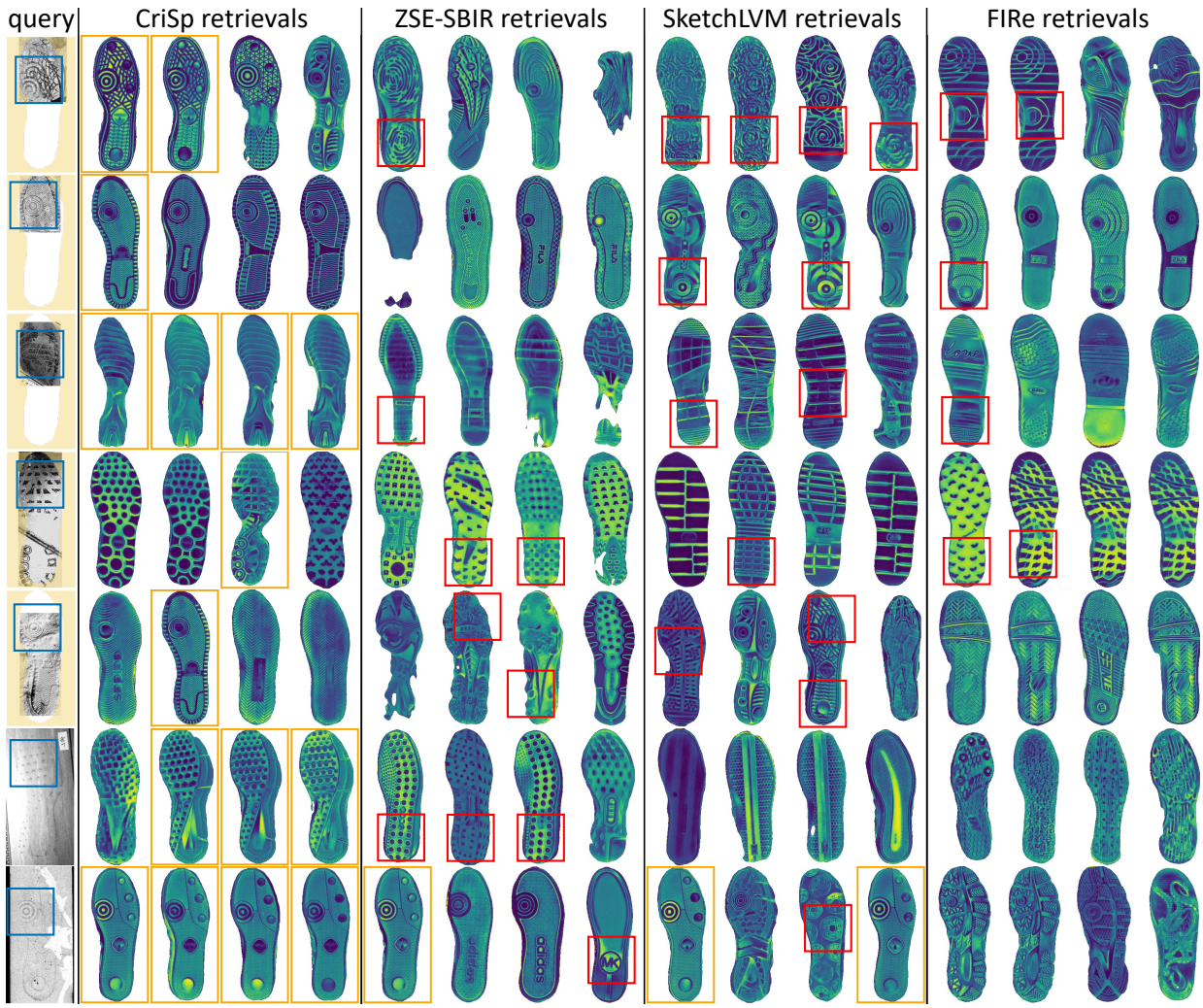

Figure 3.7: Qualitative comparison with state-of-the-art methods on val-FID (rows 1-5), val-ShoeCase (rows 6-7). We show the top 4 retrieved results. *CriSp* demonstrates the ability to localize patterns, allowing it to retrieve positive matches (highlighted in orange) much earlier than previous methods. While prior methods identify similar patterns to the query print (highlighted in blue), they cannot determine if they are from corresponding locations, as indicated by the red boxes.

3.6.1 Qualitative Results of CriSp

Figure 3.6 shows the top 10 retrievals of our method *CriSp* on the val-FID and val-ShoeCase datasets. Notable, *CriSp* can retrieve a positive match very early even when the shoeprint has significantly limited visibility or is severely degraded. These retrievals show how *CriSp* effectively matches distinctive patterns from corresponding regions of the tread. Additionally, Fig. 3.7 shows

Table 3.1: Benchmarking on real crime-scene shoeprints from val-FID. We use hit@100 and mAP@100 as our metrics and compare performance with prior methods trained on our dataset both with and without our data augmentation (see details in Sec. 3.5), which simulates crime-scene shoeprints from clean, fully-visible prints provided in the training data. Since MCNCC [48] uses features from a pretrained network, it cannot be fine-tuned on our data. Clearly, all other prior methods benefit greatly from using our data augmentation technique. Moreover, *CriSp* significantly outperforms all prior methods on both metrics, even when they are trained with our data augmentation.

method		w/o our data aug		w/ our data aug	
		hit@100	mAP@100	hit@100	mAP@100
IJCV'19	MCNCC [48]	0.0849	0.0018	-	-
NeurIPS'20	SupCon [47]	0.0472	0.0020	0.0755	0.0096
ICLR'21	FIRe [98]	0.1132	0.0014	0.2075	0.0398
CVPR'23	SketchLVM [76]	0.0849	0.0066	0.1981	0.0384
CVPR'23	ZSE-SBIR [60]	0.0943	0.0065	0.4528	0.1412
	CriSp	0.0754	0.0174	**0.5472**	**0.2071**

a comparison with related methods fine-tuned on our dataset. Clearly, *CriSp* performs significantly better at retrieving positive matches early. The supplement has further visualizations.

3.6.2 Comparison with State-of-the-art

CriSp consistently outperforms previous methods across most validation examples (details in the supplement). Table 3.1 and 3.2 list comparisons on our two evaluation metrics introduced in Sec. 3.3.2. We analyze these results below.

Comparison with shoeprint matching. MCNCC [48] employs features from pretrained networks on ImageNet for automated shoeprint matching. However, leveraging learning on shoeprint-specific data, *CriSp* exhibits superior performance on both val-FID (see Tab. 3.1) and val-ShoeCase (see Tab. 3.2). Although MCNCC proposes to use clean shoeprint impressions as the reference database to match with, we use tread depth maps to be consistent with other methods and to achieve enhanced results. The supplement has details.

Table 3.2: Benchmarking on simulated crime-scene prints from val-ShoeCase, which includes prints made by blood and dust. We use hit@100 and ma@100 as our metrics and *CriSp* performs the best across both metrics and print categories. Notably, all prior methods have been fine-tuned on our dataset using our data augmentation technique, as they perform poorly otherwise (cf. Tab. 3.1).

method	ShoeCase-blood		ShoeCase-dust	
	hit@100	mAP@100	hit@100	mAP@100
MCNCC [48]	0.0000	0.0000	0.0000	0.0000
SupCon [47]	0.0000	0.0000	0.0000	0.0000
FIRe [98]	0.3896	0.0275	0.8194	0.3779
SketchLVM [76]	0.6623	0.1058	0.5972	0.2696
ZSE-SBIR [60]	**0.8052**	0.1849	**0.9444**	0.4063
CriSp	**0.8052**	**0.4355**	**0.9444**	**0.6792**

Comparison with image retrieval. Table 3.1 and 3.2 demonstrate how our *CriSp* consistently outperforms state-of-the-art methods in image retrieval (SupCon [47], FIRe [98], SketchLVM [76], ZSE-SBIR [60]). We fine-tune these methods on our training data containing tread depth maps and clean, fully-visible shoeprints. Additionally, we use our data augmentation module Aug to simulate crime-scene shoeprints while training prior methods as the wide domain gap between crime-scene prints and the training data causes them to perform poorly otherwise (cf. Tab. 3.1). Even when prior methods use our data augmentation, *CriSp* significantly outperforms them on both val-FID (Tab. 3.1) and val-ShoeCase (Tab. 3.2). The ablation study (Tab. 3.5) shows that our spatial feature masking technique greatly improves the performance. Qualitative comparison on both validation sets in Fig. 3.7 also confirm that *CriSp* is better able to match shoeprint patterns to corresponding locations on tread depth maps, thus making positive retrievals early. This is reflected by our mAP@100 values when compared to prior methods on both validation sets (Tab. 3.1 and 3.2).

Scalability. In practice, when dealing with a large reference database, scalability becomes crucial. Unlike our closest competitor ZSE-SBIR [60], which necessitates the recomputation of all database features for each query, *CriSp* offers a scalable solution. It can precompute spatial database features

Table 3.3: Testing database image configurations. The hit@100 and mAP@100 values for FID-clean shoeprints indicate that using only tread depth as the database image configuration yields the best performance. Results for FID-crime are not reported in this experiment as we do not simulate crime-scene prints.

Database config.			FID-clean	
RGB	depth	print	hit@100	mAP@100
✓			0.195	0.066
	✓		**0.512**	**0.203**
		✓	0.171	0.015
✓	✓	✓	0.293	0.057

and efficiently perform feature masking and cosine similarity calculations for each query, enabling rapid retrieval even with extensive reference databases.

Simulating partial print. Retrievals by prior methods on partial shoeprints in Fig. 3.7 reveal instances of poorly segmented tread depth maps, where significant portions of the tread pattern have been erased. This raises the question of whether prior methods would exhibit improved performance if trained with masks simulating partial prints. However, it's worth noting that prior methods perform better when trained without such masks, as detailed in the supplement.

Val-FID vs val-ShoeCase. Methods show a wider variation in performance on Val-ShoeCase than val-FID. This discrepancy arises from the fact that val-FID contains the diversity of real crime-scene shoeprints, while val-ShoeCase systematically simulates crime-scene prints. Additionally, val-ShoeCase contains prints from shoe models with only two unique tread patterns while val-FID contains prints from 41 unique tread patterns (cf. Sec. 3.4.2).

3.6.3 Design Choices and Ablation Study

We conduct a study of our design choices by training a ResNet50 with a supervised contrastive loss and then sequentially adding modules to investigate their performance impact. Specifically, we analyze database image configurations, data augmentation techniques, and spatial feature masking.

Table 3.4: Effect of our data augmentation. We train a ResNet50 with our data augmentation and report hit@100 and mAP@100 values for FID-crime shoerpints. Our results confirm that each component of our data augmentation (visualized in Fig. 3.5) individually improves retrieval results and performs best when used together.

Data augmentation			FID-crime	
occlusion	erasure	noise	hit@100	mAP@100
			0.009	0.0000
✓			0.019	0.0003
	✓		0.075	0.0098
		✓	0.170	0.0241
✓	✓	✓	**0.226**	**0.0520**

Database image configuration. We start by testing the effectiveness of different types of database image configurations (RGB tread images, depth, and print). Our analysis shows that depth is the most relevant and informative modality, yielding the best results when used alone (Tab. 3.3). Print can be derived from depth by thresholding [81] and the extra information in rgb tread images (lighting and albedo) can be distracting.

Data augmentation. Next, we test the effectiveness of each component of our data augmentation technique. Table 3.4 shows that all 3 components contribute to improved performance and work best when used together, bringing our hit@100 and mAP@100 on FID-crime to (0.226, 0.0520) from (0.009, 0.000).

Spatial features and feature masking. With our data augmentation in place, we study the effect of spatial feature masking, which helps *CriSp* match query print patterns to the relevant spatial locations of the database tread depth maps. Table 3.5 shows the influence of using spatial features and feature masking. Our findings indicate that spatial features, feature masking, and query image masking during training all contribute greatly to improving performance.

Table 3.5: Effect of spatial features and feature masking. We validate the effect of using spatial features and applying feature masking on both our encoder Enc, which incorporates spatial features during training, and on a pretrained ResNet50 trained with our data augmentation (cf. Tab. 3.4). For ResNet50, which does not utilize spatial features during training, we obtain spatial features by removing the last pooling operation. We present results for FID-crime shoeprints from val-FID using hit@100 and mAP@100 metrics. Using spatial features from a pretrained ResNet50 boosts retrieval performance. Additionally, masking the spatial features improves performance further for both the ResNet50 and our Enc. Furthermore, adding query print masking during training further boosts performance to hit@100=0.5472 and mAP@100=0.2071.

encoder	train w/ spatial feat.	spatial features	mask features	mask query print	FID-crime	
					hit@100	mAP@100
ResNet50					0.2264	0.0520
ResNet50		✓			0.3585	0.0863
ResNet50		✓	✓		0.4245	0.1212
Enc	✓	✓			0.3774	0.1137
Enc	✓	✓	✓		0.4528	0.1765
Enc	✓	✓	✓	✓	**0.5472**	**0.2071**

3.7 Discussions and Conclusions

Limitations. While *CriSp* significantly outperforms prior methods on this problem, it still has some limitations. We use CNNs since it is straightforward to localize pattern matching to corresponding locations on images by applying spatial feature masking. However, pairing localization techniques with more sophisticated models such as vision transformers is expected to perform better. We also assume that the crime-scene shoeprints are manually aligned before queries are made. Methods that do not require this step are easier to use.

Potential negative impact. We introduce a method to aid in forensic investigations. However, if investigators tend to rely *solely* on our retrievals for crime-scene shoeprint identification, then criminals wearing shoe models that are not well-represented by *CriSp* would become harder to identify. We have to always consider the possibility that the shoe model that left a crime-scene impression may not be present in the top retrievals.

Conclusion. In this chapter, we propose a method to retrieve and rank the closest matches to crime-scene shoeprints from a database of shoe tread images. This is a socially important problem and helps forensic investigations. We introduce a way to learn from large-scale data and propose a spatial feature masking method to localize the search for patterns over the shoe tread. Our method consistently outperforms the state-of-the-art on both image retrieval and crime-scene shoeprint matching methods on our two validation sets that we reprocess from the widely used FID and more recent ShoeCase datasets. Future explorations can investigate using architectures like vision transformers and extend the problem to when alignment is not guaranteed.

Chapter 4

Conclusion and Future Directions

In this dissertation, we tackle a significant societal challenge by contributing to the forensic investigation of crime scene evidence, particularly focusing on shoeprints, which are a prevalent form of evidence encountered in criminal investigations. While shoeprints may not possess the individualized identification capabilities of biometric samples like blood or hair, they play a crucial role in narrowing down potential suspects and providing valuable leads in criminal cases, especially considering the decline in the availability of other biometric evidence at crime scenes due to increasing awareness of their forensic significance [100].

The forensic analysis of shoeprints encompasses both class and acquired characteristics. Our primary focus lies in identifying the class characteristics of shoeprints, such as brand, model, and size, rather than examining acquired traits like cuts, scratches, or wear patterns that develop on a shoe over time.

To address this challenge comprehensively, we propose a holistic approach to retrieve shoe models that best match query crime-scene shoeprints. Our methodology involves automating the generation of a large-scale reference database to which crime scene shoeprints can be matched. Subsequently, we train a retrieval network to identify the shoe models from this extensive reference

database that best match the query crime-scene shoeprints. We have also generated a synthetic dataset of tread images, with corresponding intrinsics (depth, albedo, normal, and lighting), and real datasets for training and evaluation purposes. These datasets serve as valuable resources for future research endeavors. Through extensive experimentation, we demonstrate the superiority of both our depth predictor and retrieval method over existing state-of-the-art approaches across multiple datasets, affirming the effectiveness and reliability of our proposed methodology.

4.1 Future Directions

Although our approach demonstrates significant advancements over the current state-of-the-art, there are several steps remaining before it can be seamlessly integrated into professional forensic investigations. Below, we delineate the necessary actions to transition our approach into a deployable tool for forensic labs, along with suggestions for future enhancements to further refine our methodology.

4.1.1 Application with a User Interface

While our research presents a promising approach for retrieving shoe models, its practical utility hinges on its integration into an application and packaging it as a functional tool. This application would necessitate a user-friendly interface enabling investigators to upload crime scene photos and modify them as necessary through operations such as translation, rotation, and scaling. Additionally, the interface should allow users to specify a mask indicating visible portions of the print.

To address this requirement, we have developed an application [2] where users can upload a crime scene shoeprint and an accompanying mask to retrieve the closest matching shoe models from a reference database compiled by downloading shoe tread images advertised by online retailers (detailed in Section 3.4). The retrieved shoes are ranked, and the corresponding brand name,

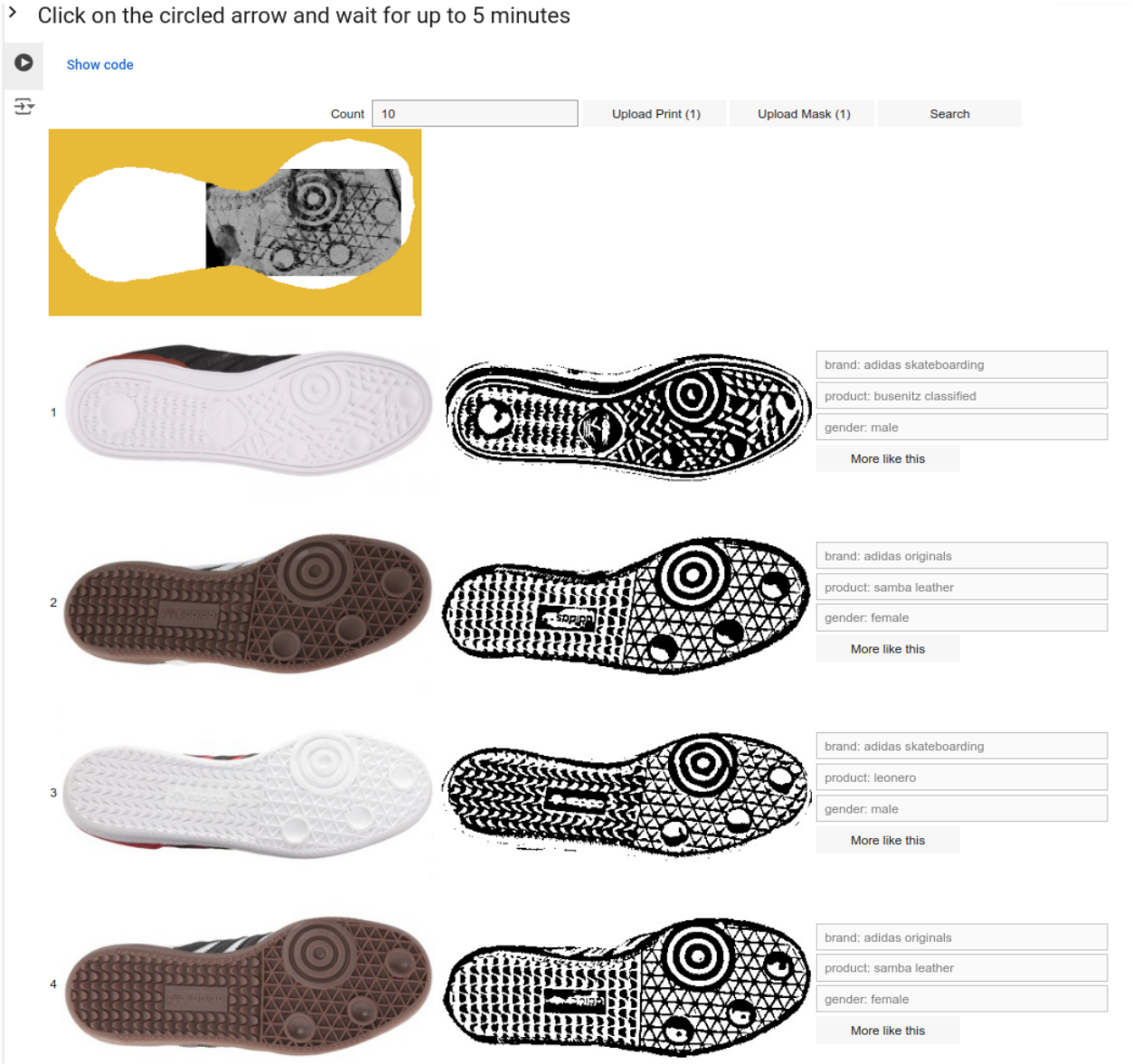

Figure 4.1: An application for retrieving the best matching shoe models to a crime-scene shoeprint. We provide a tool [2] where you can upload a crime-scene shoeprint and a corresponding mask and retrieve the best matching shoe models from a large-scale reference database of shoes created from crawling online retail stores (see details in Section 3.4). This screenshot shows an example crime-scene shoeprint queried on our tool and the retrieved results. The results and ranked and the brand name, product name and intended gender for the shoe models are shown for each result.

product name, and intended gender for the shoe are displayed. The application is packaged as a Jupyter Notebook compatible with Google Colab, ensuring accessibility across various computing platforms. There is no setup required, as the notebook automatically downloads our model and reference database in the background. A screenshot of our tool is presented in Figure 4.1.

While the application is a step forward, further enhancements are necessary to maximize its utility. This includes incorporating image modification capabilities into the interface, as discussed earlier. Moreover, the application should connect to a dynamic reference database that remains updated over time, unlike our static reference database. Such an advanced tool may warrant its own website and server for hosting purposes.

4.1.2 Reference Database

Retrieval tools developed for forensic applications necessitate the maintenance of an up-to-date reference database. A mechanism should be established to consistently gather data from the web and process it through tasks such as global alignment, depth prediction, and print prediction. Regular monitoring is essential to ensure the system remains operational over time. Although we developed a web crawler and conducted image processing steps to compile our reference database, we ceased running it once we had gathered sufficient data for our research objectives. However, for the utilization of our method by forensic investigators, it is imperative that this software remains operational and is maintained accordingly.

4.1.3 Legal Issues Regarding Use of Online Retail Images

We have downloaded several thousand images and used them for academic research. However, to transition this to a commercial product, certain legal considerations must be addressed. Specifically, we need to obtain licenses to use the tread images downloaded from online retail stores.

4.1.4 Statistical Confidence

The shoe model corresponding to a crime-scene shoeprint may not always be present in the reference database we compile from online retail stores. Therefore, it is crucial to identify when a good match is not available within our reference database. Our current retrieval method does not account for the statistical confidence of the retrieved shoe models. Future work can include a study of statistical confidence scores, ensuring they can indicate the confidence level of their matches and point out if the shoe model we are looking for is likely not present in the reference database.

4.1.5 Model Architectures

We utilize a variation of the ResNet50 model architecture for our retrieval network because it allows for easy specification of spatial features. However, more modern architectures, such as vision transformers, are expected to outperform our current approach when combined with localization techniques. Similarly, our depth predictor is based on a classic UNet with skip connections. Employing more modern model architectures for the depth predictor could also improve performance.

4.1.6 Shoe Size

Our current methodology does not account for shoe size. Tread images from online retail stores typically lack size labels or scale references. Retailers provide a representative image for a shoe model without specifying the size that was photographed and do not offer different images for each shoe size. Consequently, while we may know the scale of crime-scene shoeprints, this class characteristic is missing in the tread images (or corresponding depth map predictions) from online retail stores used for matching crime-scene shoeprints.

4.1.7 Training and Testing Datasets

A neural network's performance is highly dependent on the quality of the data it is trained on, making it essential to have robust training and testing datasets. Our synthetic dataset for training the depth predictor comprises various hypothetical shoe tread patterns that mimic real ones. However, there is room for enhancement by making these synthetic tread images more realistic. Here are some ways to improve the dataset:

- **Incorporate Complex Materials**: Including complex materials for shoe treads, such as shiny or translucent materials, will enrich the synthetic dataset. Many real shoes are made of such materials, and training with this data will enable the depth predictor to make more accurate predictions on real shoes.
- **Enhance Tread Shapes**: Currently, we use 2D shoeprints to create 3D depth maps, resulting in tread shapes that are not entirely representative of real shoes. We can achieve closer imitation by scanning the shapes of actual shoes and incorporating these 3D shapes into our synthetic treads.

In addition to enhancing our synthetic data, we can improve the real dataset used for training our retrieval network. Currently, we use Stock Keeping Units (SKUs) as our shoe model labels. SKUs are numbers used by online retailers to manage stock levels. While SKUs are generally good indicators of shoe models, they are not always accurate. For instance, the same shoe model can sometimes have different SKU numbers. This issue is illustrated in Figure 4.1, where the retrieved results have different SKUs, yet the second and fourth results both correspond to the Adidas Skateboarding, Samba Leather shoe. Developing an automated mechanism to consolidate such examples would be beneficial. Additionally, exploring other data sources for training that do not have these limitations could also be advantageous.

4.1.8 Alignment

Our current method assumes that investigators will manually align crime-scene shoeprints before querying them against the reference database. However, methods that eliminate this alignment step would be more user-friendly. Furthermore, crime-scene shoeprints can often be highly degraded or severely occluded, making it challenging to determine which part of the shoe the print is from. This complicates the task of placing the shoeprint on a ”shoe outline” (as described in Section 3.4). Therefore, future work can focus on extending our approach to handle cases where alignment is not guaranteed.

Bibliography

[1] 6pm. http://www.6pm.com.

[2] A tool to identify shoe models from crime-scene shoeprints. https://colab.research.google.com/github/Samia067/crime-scene-shoeprint-matching/blob/main/image_search.ipynb.

[3] Zappos. http://www.zappos.com.

[4] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2021.

[5] G. AlGarni and M. Hamiane. A novel technique for automatic shoeprint image retrieval. *Forensic science international*, 181(1-3):10–14, 2008.

[6] H. A. Alhaija, S. K. Mustikovela, J. Thies, V. Jampani, M. Nießner, A. Geiger, and C. Rother. Intrinsic autoencoders for joint neural rendering and intrinsic image decomposition, 2021.

[7] S. Alizadeh and C. Kose. Automatic retrieval of shoeprint images using blocked sparse representation. *Forensic science international*, 277:103–114, 2017.

[8] S. Almaadeed, A. Bouridane, D. Crookes, and O. Nibouche. Partial shoeprint retrieval using multiple point-of-interest detectors and sift descriptors. *Integrated Computer-Aided Engineering*, 22(1):41–58, 2015.

[9] R. Alp Guler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6799–6808, 2017.

[10] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[11] A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015.

[12] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 584–599. Springer, 2014.

[13] M. H. Baig and L. Torresani. Coupled depth learning. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2016.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[15] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 1000–1006. IEEE, 1997.

[16] W. J. Bodziak. *Footwear impression evidence: detection, recovery, and examination*. CRC Press, 2017.

[17] A. Bouridane, A. Alexander, M. Nibouche, and D. Crookes. Application of fractals to the detection and classification of shoeprints. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 474–477. IEEE, 2000.

[18] R. Bowen and J. Schneider. Forensic databases: paint, shoe prints, and beyond. *NIJ Journal*, 258:34–38, 2007.

[19] B. Cao, A. Araujo, and J. Sim. Unifying deep local and global features for image search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 726–743. Springer, 2020.

[20] P. N. Chowdhury, A. K. Bhunia, A. Sain, S. Koley, T. Xiang, and Y.-Z. Song. Scenetrilogy: On human scene-sketch and its complementarity with photo and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10972–10983, 2023.

[21] A. K. Datta, H. C. Lee, R. Ramotowski, and R. Gaensslen. *Advances in fingerprint technology*. CRC press, 2001.

[22] P. De Chazal, J. Flynn, and R. 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.

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

[24] S. Dey, P. Riba, A. Dutta, J. Llados, and Y.-Z. Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2179–2188, 2019.

[25] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[26] A. Dutta and Z. Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5089–5098, 2019.

[27] E. N. Efthimiadis. Query expansion. *Annual review of information science and technology (ARIST)*, 31:121–87, 1996.

[28] D. A. Forsyth and J. J. Rock. Intrinsic image decomposition using paradigms. *CoRR*, abs/2011.10512, 2020.

[29] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988.

[30] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[31] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

[32] R. Furukawa, R. Sagawa, and H. Kawasaki. Depth estimation using structured light flow–analysis of projected pattern flow on an object's surface. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4640–4648, 2017.

[33] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[34] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[35] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 241–257. Springer, 2016.

[36] M. Gueham, A. Bouridane, and D. Crookes. Automatic recognition of partial shoeprints based on phase-only correlation. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV–441. IEEE, 2007.

[37] M. Gueham, A. Bouridane, and D. Crookes. Automatic classification of partial shoeprints using advanced correlation filters for use in forensic science. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[38] M. Gueham, A. Bouridane, D. Crookes, and O. Nibouche. Automatic recognition of shoeprints using fourier-mellin transform. In *2008 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 487–491. IEEE, 2008.

[39] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3279–3286, 2015.

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

[41] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, 10–15 Jul 2018.

[42] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005.

[43] W. Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[44] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. B. Tenenbaum. Self-supervised intrinsic image decomposition. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5938–5948, Red Hook, NY, USA, 2017. Curran Associates Inc.

[45] M.-Q. Jing, W.-J. Ho, and L.-H. Chen. A novel method for shoeprints recognition and classification. In *2009 International conference on machine learning and cybernetics*, volume 5, pages 2846–2851. IEEE, 2009.

[46] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[47] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[48] B. Kong, J. Supancic III, D. Ramanan, and C. C. Fowlkes. Cross-domain image matching with deep feature maps. *International Journal of Computer Vision*, 127(11):1738–1750, 2019.

[49] X. Kong, C. Yang, and F. Zheng. A novel method for shoeprint recognition in crime scenes. In *Biometric Recognition: 9th Chinese Conference, CCBR 2014, Shenyang, China, November 7-9, 2014. Proceedings 9*, pages 498–505. Springer, 2014.

[50] A. Kortylewski, T. Albrecht, and T. Vetter. Unsupervised footwear impression analysis and retrieval from crime scene data. In *Computer Vision-ACCV 2014 Workshops: Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part I 12*, pages 644–658. Springer, 2015.

[51] A. Kortylewski, T. Albrecht, and T. Vetter. Unsupervised footwear impression analysis and retrieval from crime scene data. In *Computer Vision-ACCV 2014 Workshops: Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part I 12*, pages 644–658. Springer, 2015.

[52] A. Kortylewski and T. Vetter. Probabilistic compositional active basis models for robust pattern recognition. In *BMVC*, 2016.

[53] S. Krig. Interest point detector and feature descriptor survey. In *Computer vision metrics*, pages 187–246. Springer, 2016.

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

[55] Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017.

[56] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[57] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017.

[58] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020.

[59] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[60] F. Lin, M. Li, D. Li, T. Hospedales, Y.-Z. Song, and Y. Qi. Zero-shot everything sketch-based image retrieval, and in explainable style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23349–23358, 2023.

[61] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu. Transfer sparse coding for robust image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 407–414, 2013.

[62] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[63] Z. Ma, Y. Ding, S. Wen, J. Xie, Y. Jin, Z. Si, and H. Wang. Shoe-print image retrieval with multi-part weighted cnn. *IEEE Access*, 7:59728–59736, 2019.

[64] T. Nestmeyer, J.-F. Lalonde, I. Matthews, and A. Lehrmann. Learning physics-guided face relighting under directional light. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5124–5133, 2020.

[65] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

[66] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[67] K. Pang, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. In *BMVC*, pages 1–12, 2017.

[68] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[69] Y. Qian, L. Feng, Y. Song, X. Tao, and C. L. Chen. Sketch me that shoe. In *IEEE Conf. Comput. Vision and Pattern Recognit.(CVPR)*, pages 799–807, 2016.

[70] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 3–20. Springer, 2016.

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

[72] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[73] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4058–4066, 2016.

[74] I. Rida, L. Fei, H. Proença, A. Nait-Ali, and A. Hadid. Forensic shoe-print identification: a brief survey. *arXiv preprint arXiv:1901.01431*, 2019.

[75] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5506–5514, 2016.

[76] A. Sain, A. K. Bhunia, P. N. Chowdhury, S. Koley, T. Xiang, and Y.-Z. Song. Clip for all things zero-shot sketch-based image retrieval, fine-grained or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2765–2775, 2023.

[77] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 2005.

[78] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International journal of computer vision*, 76(1):53–69, 2008.

[79] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. In *International Conference on Computer Vision (ICCV)*, 2019.

[80] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[81] S. Shafique, B. Kong, S. Kong, and C. Fowlkes. Creating a forensic database of shoeprints from online shoe-tread photos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 858–868, 2023.

[82] S. Shafique, S. Kong, and C. Fowlkes. Crisp: Leveraging tread depth maps for enhanced crime-scene shoeprint matching. *arXiv preprint arXiv:2404.16972*, 2024.

[83] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[84] D. Shin, Z. Ren, E. B. Sudderth, and C. C. Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2172–2182, 2019.

[85] M. B. Smith. The forensic analysis of footwear impression evidence. *Forensic Science Communications*, 2009. https://archives.fbi.gov/archives/about-us/lab/forensic-science-communications/fsc/july2009/review.

[86] J. Song, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Fine-grained image retrieval: the text/sketch input dilemma. In *BMVC*, volume 2, page 7, 2017.

[87] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 5551–5560, 2017.

[88] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[89] A. Tibben, M. McGuire, S. Renfro, and A. Carriquiry. Shoecase: A data set of mock crime scene footwear impressions. *Data in Brief*, 50:109546, 2023.

[90] G. Tolias, Y. Avrithis, and H. Jégou. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision*, 116:247–261, 2016.

[91] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[92] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[93] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[94] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[95] X. Wang, C. Zhang, Y. Wu, and Y. Shu. A manifold ranking based method using hybrid features for crime scene shoeprint retrieval. *Multimedia Tools and Applications*, 76(20):21629–21649, 2017.

[96] Z. Wang, X. Yu, M. Lu, Q. Wang, C. Qian, and F. Xu. Single image portrait relighting via explicit multiple reflectance channel modeling. *ACM Trans. Graph.*, 39(6), Nov. 2020.

[97] C.-H. Wei and C.-Y. Gwo. Alignment of core point for shoeprint analysis and retrieval. In *2014 International Conference on Information Science, Electronics and Electrical Engineering*, volume 2, pages 1069–1072. IEEE, 2014.

[98] P. Weinzaepfel, T. Lucas, D. Larlus, and Y. Kalantidis. Learning super-features for image retrieval. In *International Conference on Learning Representations*, 2021.

[99] S. Wu, A. Makadia, J. Wu, N. Snavely, R. Tucker, and A. Kanazawa. De-rendering the world's revolutionary artefacts. In *CVPR*, 2021.

[100] Y. Wu, X. Dong, G. Shi, X. Zhang, and C. Chen. Crime scene shoeprint image retrieval: A review. *Electronics*, 11(16):2487, 2022.

[101] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European conference on computer vision*, pages 842–857. Springer, 2016.

[102] Y. Yekutieli, Y. Shor, S. Wiesner, and T. Tsach. Expert assisting computerized system for evaluating the degree of certainty in 2d shoeprints. *The US Department of Justice: Washington, DC, USA*, 2012.

[103] S. K. Yelamarthi, S. K. Reddy, A. Mishra, and A. Mittal. A zero-shot framework for sketch based image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018.

[104] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[105] S. L. Yongjie Zhu, Yinda Zhang and B. Shi. Spatially-varying outdoor lighting estimation from intrinsics. In *CVPR*, 2021.

[106] Y. Yu, A. Meka, M. Elgharib, H.-P. Seidel, C. Theobalt, and W. Smith. Self-supervised outdoor scene relighting. In *European Conference on Computer Vision (ECCV)*, 2020.

[107] Y. Zhang, H. Fu, E. Dellandréa, and L. Chen. Adapting convolutional neural networks on the shoeprint retrieval for forensic use. In *Chinese Conference on Biometric Recognition*, pages 520–527. Springer, 2017.

[108] Y. Zhao, S. Kong, and C. Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15759–15768, 2021.

[109] Y. Zhao, S. Kong, D. Shin, and C. Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3330–3340, 2020.

[110] C. Zheng, T.-J. Cham, and J. Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[111] L. Zheng, Y. Yang, and Q. Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2017.

[112] H. Zhou, X. Yu, and D. Jacobs. Glosh: Global-local spherical harmonics for intrinsic image decomposition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7819–7828, 2019.

[113] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Appendix A

Creating a Forensic Database of Shoeprints from Online Shoe-Tread Photos

Outline

We propose to create a forensic database of shoeprints by leveraging shoe-tread imagery collected by online retailers. To do so, we strive to predict depth maps for these photos, thresholding whihc generates shoeprints used to match a query print (collected from a crime scene). We propose a novel method, *ShoeRinsics*, to learn depth estimation from synthetic data (along with intrinsic components) and real data (with no annotations). *SheoRinsics* incorporates synthetic-to-real domain adaptation and intrinsic image decomposition techniques to mitigate domain gaps. We validate our method with a defined evaluation protocol that measures the degree of match between predicted depth and ground-truth shoeprints (collected in a lab environment). Results convincingly demonstrate that *ShoeRinsics* remarkably outperforms state-of-the-art methods for shoe-tread depth prediction. In this supplementary document, we discuss the following topics:

- Section A.1 details the process of matching a ground-truth shoeprint to predicted depth for evaluation.
- Section A.2 shows qualitative analysis of our *ShoeRinsics*. We visualize of all predictions of *ShoeRinsics* on real-FID-val images downloaded from the internet in Section A.2.1 and compare to RIN [44] in Section A.2.2.
- Section A.3 compares performance of *ShoeRinsics* with related work for each individual image from real-val and real-FID-val.
- Section A.4 provides details on our synthetic dataset generation. The process of depth map generation is described in Section A.4.1 and the light environemnts used are visualized in Section A.4.2.
- Section A.5 describes how we photograph shoe-treads and collect their prints to create a validation set (real-val) for quantitative evaluation.
- Section A.6 details the architecture of each component of our *ShoeRinsics*.
- Section A.7 discusses the pseudo albedo generated for real shoe-tread images and compares them to the albedo predicted by *ShoeRinsics*.

Algorithm 1 Metric of Depth-Print Matching

Input: predicted depth $\hat{X}_R^d$, ground-truth shoeprint S^*, and mask m

Initialize best-matching IoU $v_{max} = 0$

determine per-pixel local depth, $d_l = \frac{\text{blurred depth}}{\text{blurred mask}}$

for $s \in [0.1, 0.11, 0.12, \ldots, 2]$ **do**

  if IoU$(\hat{X}_R^d < sd_l, S^*) > v_{max}$ **then**

    $v_{max} = \text{IoU}(\hat{X}_R^d < sd_l, S^*)$,

    set best-matching shoeprint $S_{best} = \hat{X}_R^d < sd_l$

  end if

end for

set p_{95} = value at the 95th percentile in sorted $\hat{X}_R^d$

for $t_{nc} \in [0.1p_{95}, 0.1p_{95} + 0.01, 0.1p_{95} + 0.02, \ldots, p_{95}]$ **do**

  determine shoeprint $S_{t_{nc}} = S_{best}$ AND $(\hat{X}_R^d < t_{nc})$

  if IoU$(S_{t_{nc}}, S^*) > v_{max}$ **then**

    $v_{max} = \text{IoU}(S_{t_{nc}}, S^*), \quad S_{best} = S_{t_{nc}}$

  end if

end for

set p_{05} = value at the 5th percentile in sorted $\hat{X}_R^d$

for $t_c \in [p_{05}, p_{05} + 0.1, p_{05} + 0.2, \ldots, 30p_{05}]$ **do**

  determine shoeprint $S_{t_c} = S_{best}$ OR $(\hat{X}_R^d < t_c)$

  if IoU$(S_{t_c}, S^*) > v_{max}$ **then**

    $v_{max} = \text{IoU}(S_{t_c}, S^*)$

  end if

end for

return best-matching IoU v_{max}

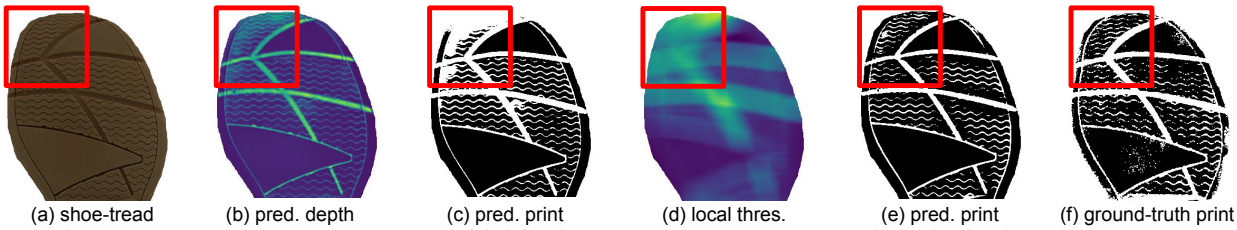

Figure A.1: Effect of adaptive threshold. Given a shoe-tread image (a), we predict its depth (b). Notice that the front of the shoe is curved up slightly (highlighted by the red boxes). Thus, a global threshold fails to capture the print properly. Compare the print prediction using a global threshold (c) with the ground-truth print (f). A solution is to use an adaptive threshold instead. As such, we first blur the predicted depth to get the local mean depth d_l. Using sd_l as the local threshold (d) where s is an appropriate constant for scaling, we get predicted print (e) which is much closer to the ground-truth (f).

A.1 Depth-Print Matching in Evaluation

We get the shoeprint prediction by thresholding the predicted depth $\hat{X}_R^d$ of a shoe-tread image. However, different thresholds can produce different predicted shoeprints. So, we develop a threshold-free metric for evaluating how well our predicted depth matches a ground-truth shoeprint. Ideally, we want to threshold the depth prediction in a way that produces a shoeprint prediction that most closely matches the ground-truth shoeprint. We summarize our method in Algorithm 1.

Global thresholding vs. adaptive thresholding. Using a global threshold for print prediction from depth prediction is troublesome since errors can creep into regions where the shoe-tread curves upwards (for example, in the front of the shoe). Fig. A.1 illustrates this scenario with a sample shoe from real-val. In such cases, although the shoe is curved upwards, it still leaves a print when someone walks wearing those shoes. This is because the weight of the person flattens out the shoe. Also, the physical motions of walking causes the curved parts to come in contact with the ground. Assuming high depth values correspond to non-contact surfaces, a portion of the shoe that curves up would have high depth values and a global threshold might incorrectly indicate that region does not leave a print. We can solve this issue by using adaptive thresholding instead.

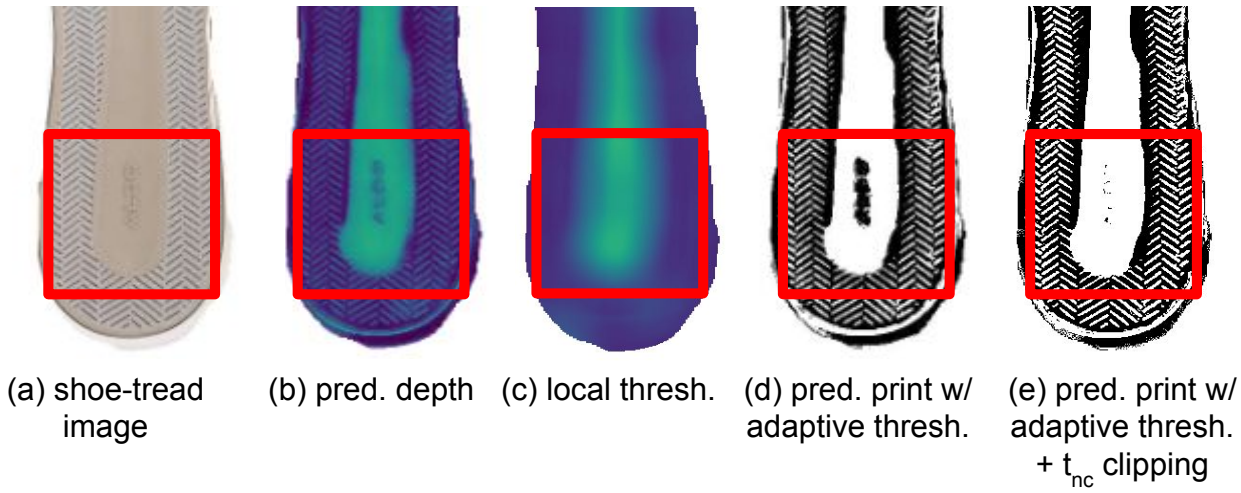

Figure A.2: Effect of non-contact threshold t_{nc}. A shoe-tread image (a), the corresponding depth prediction (b), the local threshold (c), and the predicted print prediction with adaptive threshold (d) is shown. We can see that using only the adaptive threshold can cause errors in large areas of non-contact surface as shown by the red boxes. Assume non-contact surfaces have high depth values. Although the logo is correctly predicted to have a high depth value, the local threshold also happens to be high in the region and causes adaptive thresholding to undesirably predict the logo to leave a print. To correct this, we find an appropriate non-contact threshold t_{nc} for which regions where predicted depth is greater than t_{nc} does not leave a print. The resulting print is shown in (e).

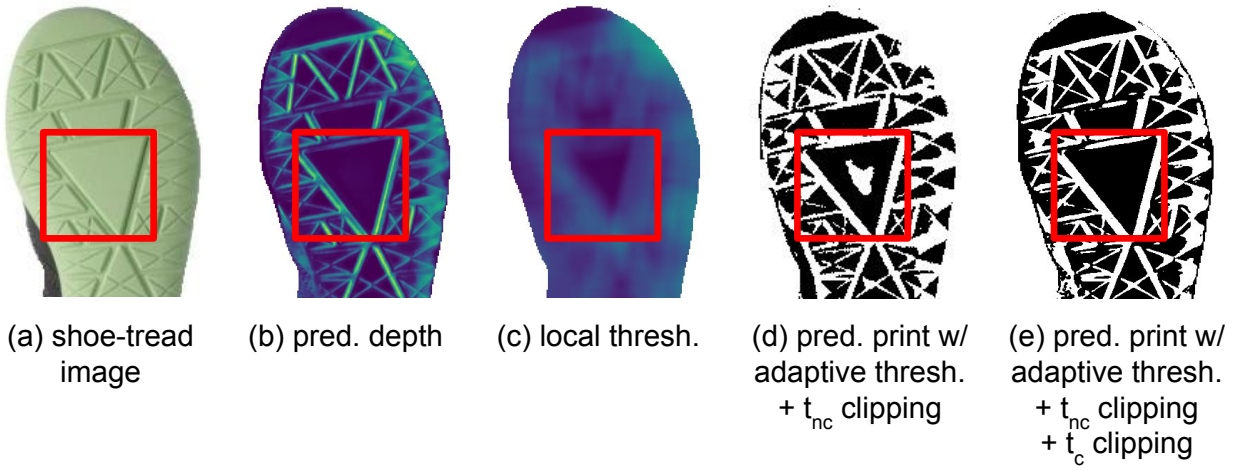

Figure A.3: Effect of contact threshold t_c. We visualize a sample shoe-tread image (a), the predicted depth (b), the local threshold (c), and the print prediction after using adaptive thresholing with t_{nc} clipping (d). Parts of a large contact surface incorrectly leaves no print as shown by the red boxes because the local threshold is very low in the region (assuming contact areas have low depth value). To correct this, we determine an appropriate contact threshold t_c such that regions where predicted depth is less than t_c always leave a print. (e) demonstrates the final result.

Details of adaptive thresholding. To perform adaptive thresholding, we first determine a per-pixel local average d_l for the predicted depth. This is achieved by blurring the predicted depth $\hat{X}_R^d$ with

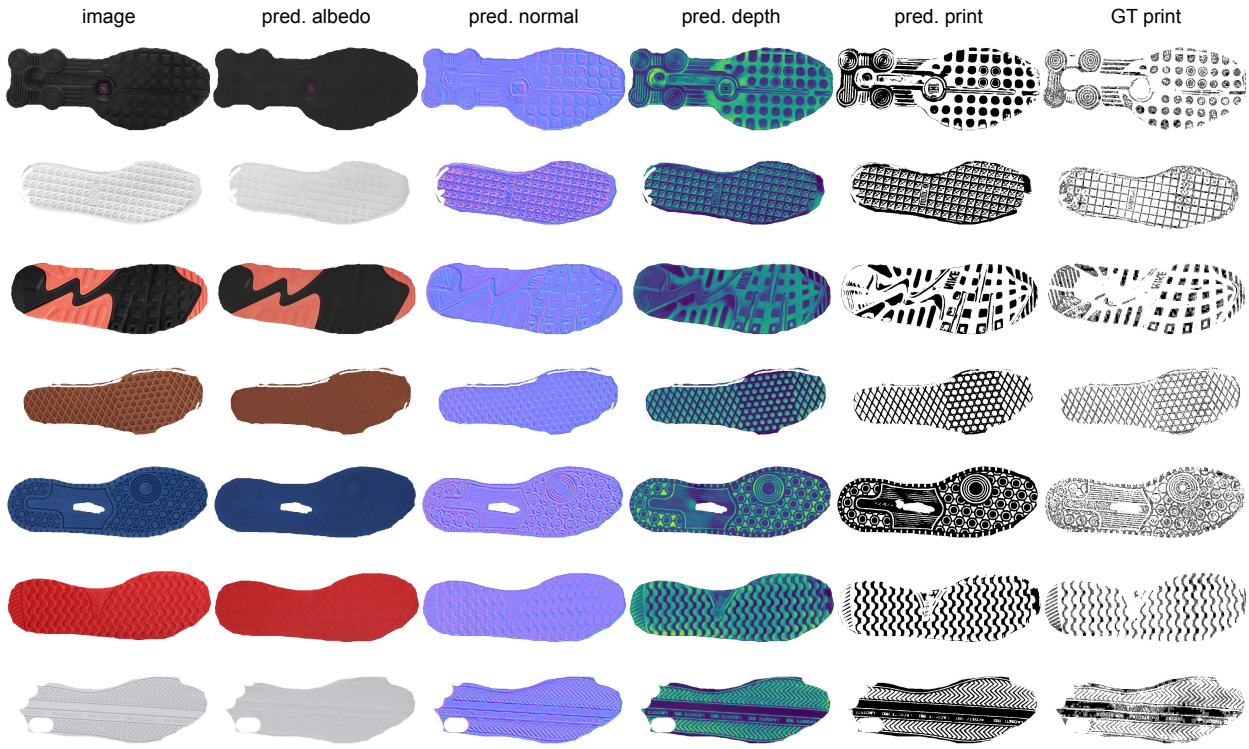

Figure A.4: Visualization of predicted shoeprints, as well as intrinsics, by our *ShoeRinsics* on real-FID-val. Real-FID-val consists of images of real shoe-treads downloaded from online retailers and corresponding ground-truth shoeprints. Visually, we can see our method works quite well w.r.t both shoeprint prediction and intrinsic decomposition (albedo, normal, and depth).

a large square kernel of size 45×45. For comparison, our shoe-tread image and predicted depth map resolution is 405×765. We note that boundaries and invalid depth values outside the mask may cause artifacts in d_l. To negate this effect, we set $d_l = \frac{d_l}{m_l}$ where m_l is the per-pixel local average for the mask computed in a similar manner.

Next, we set our best-matching shoeprint $S_{best} = \hat{X}_R^d < sd_l$ for some scalar multiplier s. Theoretically, we want to sweep over all possible values of s and find the one which gives the highest IoU between $\hat{X}_R^d < sd_l$ and the ground-truth print S^*. Practically, we sample values from range $[0.1, 2]$ at intervals of 0.01.

Threshold t_{nc} for large non-contact regions. Although, our current best-matching shoeprint estimation is good enough for most cases, it may have issues for very large non-contact surfaces. Fig. A.2 illustrates how a large non-contact region can cause the local threshold sd_l to be very

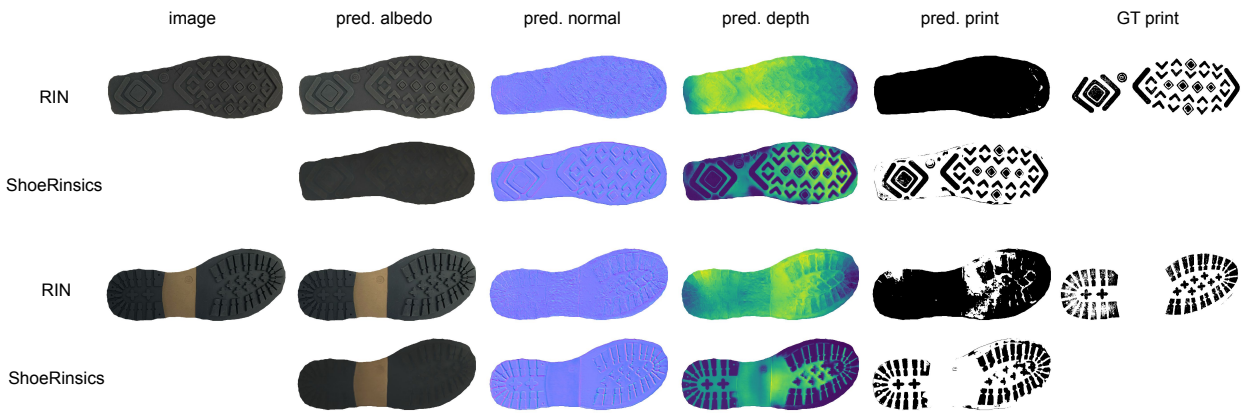

Figure A.5: Qualitative comparison between RIN [44] and *ShoeRinsics* on real-val. Along with input images and ground-truth shoeprint, we show albedo, normal, depth, and print prediction. Note that RIN does not directly produce depth predictions. We obtain them by integrating their normal predictions using the well-established Frankot Chellappa algorithm [29]. As we can see, RIN produces poor quality albedo and normal predictions, presumably because it does not explicitly perform domain adaptation. The albedo predictions retain much of the shading information and the normal predictions are noisy. The substandard normal predictions in RIN lead to unsatisfactory depth and print predictions. Comparatively, *ShoeRinsics* is able to produce more likely albedo, normal, and depth predictions and thus predict shoeprints which are much closer to the ground-truth.

high. This in turn can lead to incorrectly predicting some areas to leave a print (such as the logo in Fig. A.2). We fix this by identifying the threshold t_{nc} which gives the highest IoU between ground-truth shoeprint S^* and S_{best} AND $(\hat{X}_R^d < t_{nc})$ and update our shoeprint prediction S_{best}. We find it sufficient to determine t_{nc} by sampling values in range $[0.1p_{95}, p_{95}]$ at an interval of 0.01 where p_{95} is the 95^{th} percentile of sorted $\hat{X}_R^d$ values.

Threshold t_c for large contact regions. A similar problem and solution apply for large contact surfaces as demonstrated in Fig. A.3. In such regions, the local threshold is very low and can result in "holes" in our predicted print. Our solution is to find threshold t_c for which the IoU between ground-truth shoeprint S^* and S_{best} OR $(\hat{X}_R^d < t_c)$ is highest. We can find an adequate value for t_c by sampling numbers from range $[p_{05}, 30p_{05}]$ at intervals of 0.1 where p_{05} is the 5^{th} percentile of sorted $\hat{X}_R^d$ values.

Generic thresholds for print prediction. To determine shoeprint predictions from real images without ground-truth print, we set $s = 1$, $t_{nc} = p_{97}$, and $t_c = p_{03}$ where p_x is the x^{th} percentile of sorted $\hat{X}_R^d$ values.

A.2 Qualitative Results of *ShoeRinsics* on Real Shoe-Treads

In this section, we perform additional qualitative analysis of our *ShoeRinsics*. We visualize our albedo, normal, depth, and print predictions (Fig. A.4) as well as compare shoeprint predictions to that of RIN [44] (Fig. A.5).

A.2.1 Visualization of Predictions on Real-FID-val

One of the datasets we collected, real-FID-val, consists of images of real shoe-treads downloaded from online retailers with corresponding ground-truth shoeprints. We visualize our intrinsic predictions (albedo, normal, and depth) and compare print predictions to ground-truth shoeprints on the real-FID-val dataset in Figure A.4. We see that the intrinsic predictions are visually pleasing and the predicted print closely resembles the ground-truth shoeprint.

A.2.2 Comparison with RIN

RIN [44] learns from unlabeled real images using intrinsic image decomposition. It breaks down images into albedo, normal and light. We integrate their normal predictions to obtain a depth prediction using the well-established Frankot Chellappa algorithm [29]. Thresholding this depth prediction gives us the shoeprint prediction which we compare to the ground-truth shoeprint. In Figure A.5, we compare the albedo, normal, depth, and shoeprint predictions of RIN on real-val with that of *ShoeRinsics*. We find that RIN performs poorly on real shoes, presumably because it

Table A.1: Comparison of Intersection over Union (IoU) values achieved by *ShoeRinsics* and related work on each example in real-val. The best IoU per shoe example is written in bold and the second-best is underlined. We can see that *ShoeRinsics w/ aug* is the clear winner while *ShoeRinsics* is the second-best.

Shoe ID	RIN [44]	ADDA [93]	UDAB [33]	CyCADA [41]	ShoeRinsics	ShoeRinsics w/ aug
0001-L	25.7	29.9	34.1	38.8	41.4	**43.5**
0001-R	24.2	22.8	29.0	33.0	34.3	**36.8**
0002-L	27.8	29.3	27.8	**35.5**	29.6	32.4
0002-R	26.4	28.8	28.5	**36.9**	33.7	35.0
0003-L	20.3	24.5	30.2	32.3	37.7	**37.9**
0003-R	21.4	22.7	25.8	28.7	33.0	**33.7**
0004-L	26.3	31.0	31.0	38.0	39.6	**39.7**
0004-R	23.3	28.5	30.1	32.6	**37.5**	35.9
0005-L	29.2	50.4	56.0	59.2	**60.5**	59.9
0005-R	27.1	54.3	60.0	56.6	**62.4**	60.6
0006-L	31.5	36.2	36.3	37.7	42.2	**43.2**
0006-R	30.0	34.7	36.0	38.5	38.0	**40.4**
0007-L	19.6	19.3	19.4	17.1	22.8	**63.3**
0007-R	23.1	24.0	24.0	22.6	32.5	**42.4**
0009-L	48.8	48.8	48.9	**58.4**	55.8	49.7
0009-R	47.3	50.5	48.9	55.6	**56.1**	47.9
0010-L	52.5	56.2	54.8	56.4	**66.6**	61.0
0010-R	46.8	49.4	46.5	**53.5**	53.2	52.9
0011-L	22.4	24.6	25.0	25.0	25.0	**25.1**
0011-R	25.7	**28.9**	28.2	**28.9**	28.5	28.8
0012-L	32.1	77.5	68.2	75.8	76.2	**83.3**
0012-R	30.5	72.3	69.6	72.5	69.1	**76.9**
0013-L	24.2	**35.7**	31.7	31.5	31.2	32.8
0013-R	27.5	**40.9**	36.6	38.2	37.9	39.0
0014-L	13.6	23.9	21.2	18.8	27.7	**31.4**
0014-R	15.3	29.4	29.0	22.1	32.4	**37.8**
0015-L	24.9	**36.8**	29.8	35.0	35.3	34.4
0015-R	27.4	45.5	41.1	41.8	53.1	**53.2**
0016-L	21.7	63.4	61.1	66.0	63.9	**68.6**
0016-R	21.4	61.3	60.5	65.3	63.6	**67.8**
0017-L	24.8	47.6	47.3	56.3	55.3	**57.1**
0017-R	26.4	47.8	54.2	**60.2**	59.2	57.3
0018-L	34.8	35.3	37.3	46.4	**51.8**	50.6
0018-R	37.9	38.6	38.0	48.9	**54.1**	52.8
0019-L	66.4	69.2	72.0	73.5	71.4	**75.4**
0019-R	66.1	68.7	73.2	75.2	72.4	**76.0**
Average	30.4	41.4	41.4	44.8	46.8	**49.0**

does not explicitly perform domain adaptation. Even though our focus is on the depth prediction, our albedo and normal predictions visually look better than the predictions made by RIN. Albedo predictions from RIN retain much of the shape information. More importantly, noisy normal predictions from RIN integrate to give low quality depth predictions and thus unsatisfactory shoeprint predictions.

A.3 Further Details of Quantitative Analysis

We compare methods using our defined metric based on Intersection over Union (IoU). We analyse the IoU values for each of the shoe examples in real-val (Table A.1) and real-FID-val (Table A.2) to further demonstrate that *ShoeRinsics* outperforms the state-of-the-art domain adaptation and intrinsic image decomposition methods. We can see from the Tables that *ShoeRinsics w/ aug* performs the best, followed by *ShoeRinsics* in the second position.

A.4 Synthetic Data Preparation

To train our model, we need shoe-sole images with paired ground-truth albedo, depth, normal and light information. Publicly available datasets that include shoe objects (among other categories) [4] either do not focus on the shoe-sole and/or do not provide full decomposition into shape, albedo, and lighting. Thus, we introduce our own synthetic dataset, syn-train.

For this purpose, we synthesize depth maps, albedo maps, and lighting environments. We observe that commercial shoe tread photographs are taken under very diffuse lighting conditions where the primary variations in surface brightness are driven by global illumination effects rather than surface normal orientation (e.g., grooves appear darker). This necessitates the use of a physically-based rendering engine [43] rather than simple local shading models. We discuss details of depth map generation in Section A.4.1 and visualize the different light environment maps in Section A.4.2

A.4.1 Depth Map Generation

We use an existing shoeprint dataset [102] collected in a controlled lab environment. Sample shoeprints are displayed in Fig. A.6. We convert the 2D shoeprints to 3D depth maps by adding fake depth values to each point on the print. We generate 10-15 different depth maps from each

Table A.2: Comparison of Intersection over Union (IoU) values achieved by *ShoeRinsics* and related work on each example in real-FID-val. The best IoU per shoe example is written in bold and the second-best is underlined. We can see that *ShoeRinsics w/ aug* is the clear winner while *ShoeRinsics* is the second-best.

Shoe ID	RIN [44]	ADDA [93]	UDAB [33]	CyCADA [41]	ShoeRinsics	ShoeRinsics w/ aug
1	**33.0**	28.5	28.5	30.1	31.8	32.6
3	18.8	18.8	26.7	**35.9**	33.7	33.4
4	20.3	21.6	27.7	29.2	27.9	**29.7**
5	23.8	24.1	26.9	28.3	**29.9**	29.5
8	11.9	14.4	18.3	**21.4**	19.3	20.7
9	21.0	20.6	27.3	31.5	31.8	**32.3**
10	15.8	21.5	16.4	**23.8**	22.2	22.7
11	25.5	32.1	34.3	34.0	35.1	**36.5**
12	30.7	29.5	27.7	31.5	**32.5**	32.4
13	33.3	30.1	32.1	34.0	33.2	**34.4**
16	28.9	30.6	37.7	**52.4**	51.7	51.9
17	24.8	22.5	33.0	**35.9**	29.9	29.6
23	28.3	29.0	30.7	**32.8**	31.0	31.9
32	36.0	43.0	42.6	**47.0**	46.3	46.5
33	28.3	28.3	28.0	27.3	29.4	**30.0**
35	34.3	40.5	40.1	40.6	40.6	**41.2**
45	31.2	30.8	33.3	32.8	36.9	**37.2**
47	24.8	24.1	24.0	24.0	**24.9**	**24.9**
53	11.7	11.8	11.7	12.1	**13.7**	13.1
54	22.0	22.0	22.0	22.1	**29.1**	29.0
55	30.6	28.7	29.4	30.1	31.6	**31.8**
56	19.0	**19.3**	19.2	19.1	19.0	19.0
62	33.6	33.9	36.3	36.8	**38.1**	**38.1**
72	28.2	28.1	28.2	**29.0**	28.2	28.2
74	32.8	34.1	35.5	34.7	36.3	**36.8**
82	44.2	36.8	41.3	45.5	45.3	**45.7**
1040	42.0	37.8	40.3	45.4	46.0	**47.1**
1041	27.6	36.7	35.5	35.3	38.0	**38.1**
1044	19.5	20.7	20.6	21.5	23.9	**24.3**
1047	29.1	29.2	30.2	31.2	**31.4**	**31.4**
1048	23.5	27.7	28.5	28.1	28.0	**28.6**
1049	26.7	21.6	25.3	27.7	27.8	**28.2**
1050	26.4	25.4	26.4	26.1	**26.5**	26.4
1058	24.2	36.5	**38.8**	35.2	36.7	38.0
1062	19.0	21.0	23.5	25.6	**29.1**	28.6
1064	18.0	20.7	23.4	21.6	24.4	**24.6**
1071	11.7	16.5	19.2	**19.8**	18.3	18.3
1076	24.3	30.0	30.1	33.6	**34.5**	33.6
1079	26.4	28.4	29.4	29.3	31.3	**31.4**
1088	29.2	27.4	29.9	33.5	34.0	**34.2**
1095	26.8	29.7	28.7	37.7	38.1	**41.4**
Average	26.0	27.2	29.0	31.1	31.6	**32.0**

of the 387 shoeprints available in [102]. Fig. A.7 highlights major steps in depth map generation from shoeprint images. Details of depth map generation is provided below.

Removing noise. Raw shoeprint data is noisy (as shown in Fig A.6). We employ two tricks to filter noise. First, we compute a mask for the shoeprint to remove notes and dirt in the background from consideration. Given that the shoeprints are orange colored and the background does not

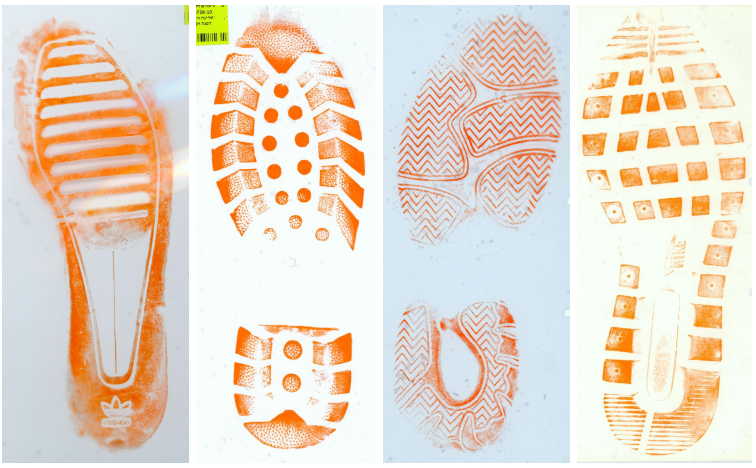

Figure A.6: Examples from an existing shoeprint dataset [102]. We use these prints as a starting point for synthetic depth map generation. Note that even though these are shoeprints collected under a controlled lab environment, the images are still quite noisy. This necessitates some preprocessing before these can be used for synthetic depth map generation.

contain any of that color, we determine the mask using the concave hull of the orange colored regions. Second, we filter noise by applying a Gaussian blur followed by a sigmoid function on the gray-scale shoeprint.

Adding a realistic touch. At this stage, our depth map mainly consists of the two extreme values representing contact and non-contact surfaces. To incorporate some texture, we add a moderated amount of high frequency details (obtained from subtracting the blurred depth from the original gray-scale shoeprint). Next, we optionally add slanted bevels to our depth map to make the tread blocks look more natural. We further add a local curvature to the non-contact surfaces to give them some dimension. Essentially, we square the euclidean distance transform of the depth image and add the smoothed out result to our depth map. Finally, we also add a global curvature along the edge of the shoe-tread to attain the natural upward curvature that is common in many shoes.

A.4.2 Visualization of Light Environments

We display the different light environments in our synthetic dataset (syn-train) in Figure A.8 and also provide a video to better visualize the lighting effects. We have a total of 17 different light

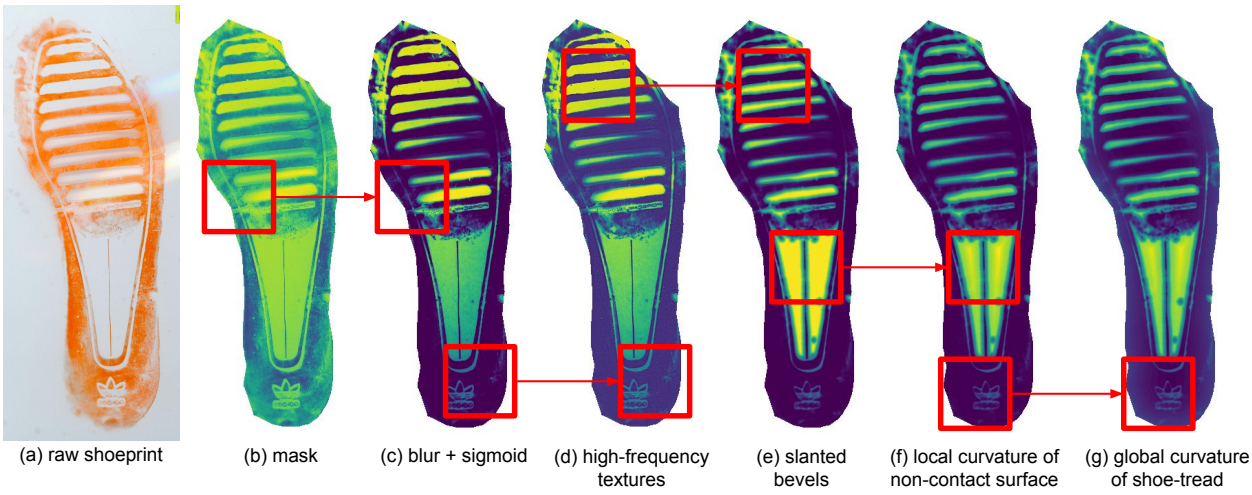

Figure A.7: We illustrate major steps in depth map generation with an example. We first filter noise in a shoeprint image (a) by masking out the background (b) and applying a Gaussian blur followed by a sigmoid function (c) on the shoeprint image. Then, we add in some moderated amount high frequency details from the shoeprint image as textures (d). To make our depth maps more realistic, we optionally add slanted bevels (e), local curvatures for non-contact surfaces (f), and a global curvature along the edge of the shoe-tread (g).

environments in our dataset. One consists of diffuse white light. Eight light configurations consist of a white light bulb providing directional light (from 8 different directions) in addition to the diffuse white light. The other eight light configurations consists of two white light bulbs at a 120° angle to each other in addition to the diffuse white light. For each light configuration, we visualize a shiny sphere in place of the shoe demonstrating the light placements. Additionally, we render an example synthetic shoe under all the different light conditions to show the effect of each of them. We see different light environments create different shadows on on the shoe-tread blocks.

A.5 Real Data Preparation for Evaluation

To quantitatively evaluate and compare methods, create real-val which consists of paired shoe-tread images and ground-truth prints for real shoes.

Photographing shoe-treads. Real-val contains new-athlectic, used-athletic, and new-formal shoes. The new shoes are collected from thrift stores which often sell new or very lightly used shoes. The

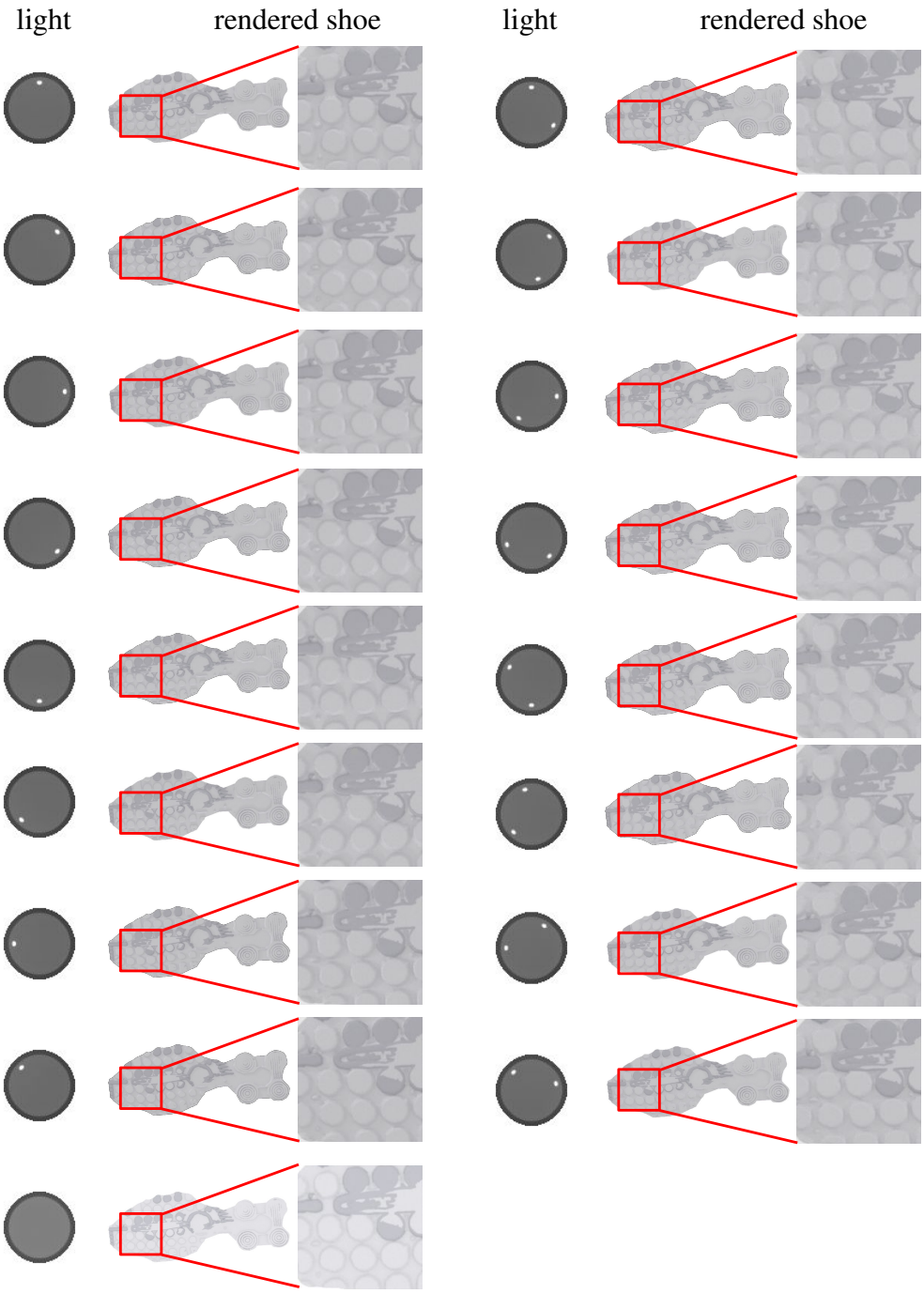

Figure A.8: Visualization of the 17 different light types in our synthetic dataset (syn-train). We show a shiny sphere representing the light in the environment and a shoe rendered under that light condition. Our light environments consist of diffuse white light in addition to 0 to 2 light bulbs for directional light. Different light sources produce different shadows on the shoe-tread blocks. Please refer to the attached video for a better visualization.

used shoes are worn-out athletic shoes donated by volunteers. We first clean all the shoe-treads with soap and water and let them dry. Next, we photograph the shoe-treads in a brightly lit environment similar to that of a professional photography setting. We put together 5 square light panels to create a light box and place the shoe on a holder inside the light box. We also illuminate the shoes using a ring light on top.

Preparing ground-truth shoeprint. After photographing the shoes we proceed to collecting their prints. We use a process called *block printing technique* which is widely used in forensics to collect lab shoeprint impressions [16]. With the shoe resting on the holder, we paint the shoe-tread with a thin layer of relief ink using a roller. Forgoing the roller and simply using a paint brush would cause ink blobs to get stuck in the nooks and crannies of the shoe-tread leading to blotchy prints. While the ink is still wet, we quickly press a slightly absorbent white paper onto the shoe-tread using a roller. The use of the roller distributes pressure throughout the paper and thus produces more uniform prints. We collect 2-3 sets of prints for each shoe, each time painting the shoe with a new layer of ink. Notice how these individual prints are not identical and contain areas of uneven coverage. To get a smoother result, we align all the prints to the shoe-tread image (using thin-plate spline [25] and point correspondences between the shoe-tread image and the collected prints) and average them. The average is a more complete and evenly colored print. Finally, the average print is thresholded to get our binary ground-truth shoeprint.

A.6 Further Implementation Details

Decomposer $\mathcal{F}$. Our decomposer consists of a classic encoder-decoder structure with skip connections. We use separate decoders for albedo, normal, depth, and light predictions. All of the encoded input is passed to each of these decoders. The light decoder consists of residual blocks followed by a final convolution layer which outputs 17 numbers representing the probability of predicting the 17 light types in our synthetic training dataset. We use the output of the second last

layer of the albedo, normal, and depth decoder as the corresponding features. For light features, we use the 17 light probabilities.

Renderer $\mathcal{R}$. The renderer has a mirrored structure as the decomposer. It has separate encoders for albedo, depth, normal, and light. The light encoder takes in a one-hot array representing the light configuration. The encoded information from each of the encoders is concatenated and passed to the decoder which predicts the synthetic or real shoe-tread image.

Connecting the decomposer and renderer. When passing decomposer outputs to the renderer in our main pipeline, we ensure that the decomposer outputs look similar to the synthetic albedo, depth, normal, and light used to train the renderer. We set the background (i.e., parts outside the shoe-tread) pixel values to 1 in the albedo, depth, and normal predictions. We also use the Hard-Gumble trick to represent the light predictions as one-hot vectors instead of fractional probabilities for the renderer. This ensures that a path exists for gradient back-propagation through the light decoder while providing a one-hot representation for the light probabilities.

Image translators $\mathcal{G}_{S\to R}$ and $\mathcal{G}_{R\to S}$. We use a ResNet backbone for the image translators. The two generators have the exact same structure. They consist of 2 convolution layers with stride 2, followed by 9 residual blocks, and finally 2 convolution layers coupled with nearest 2D upsampling layers with a scale factor of 2. The convolution layers and residual blocks in the generators are interspersed with batch normalization and the leakyReLU activation function.

Image discriminators $\mathcal{D}_S$ and $\mathcal{D}_R$. The discriminators used to learn image translation are PatchGANs and consist of 4 convolution layers with stride 2 followed by 2 convolution layers with stride 1. Similar to the image translators, the discriminators also have batch normalization and the leakyReLU activation function interspersed among the convolution layers and residual blocks.

Feature discriminator $\mathcal{D}_{feat}$. The discriminator for feature alignment takes in the concatenation of the albedo, normal, and depth features as one input, and the light features as a second input. These features are processed in two separate branches and the results are concatenated in the final

output. Each of these branches consist of 3 convolution layers. The branch for the albedo, normal, and depth features uses a kernel size of 3 (to encode some context), while the branch for light features use a kernel size of 1.

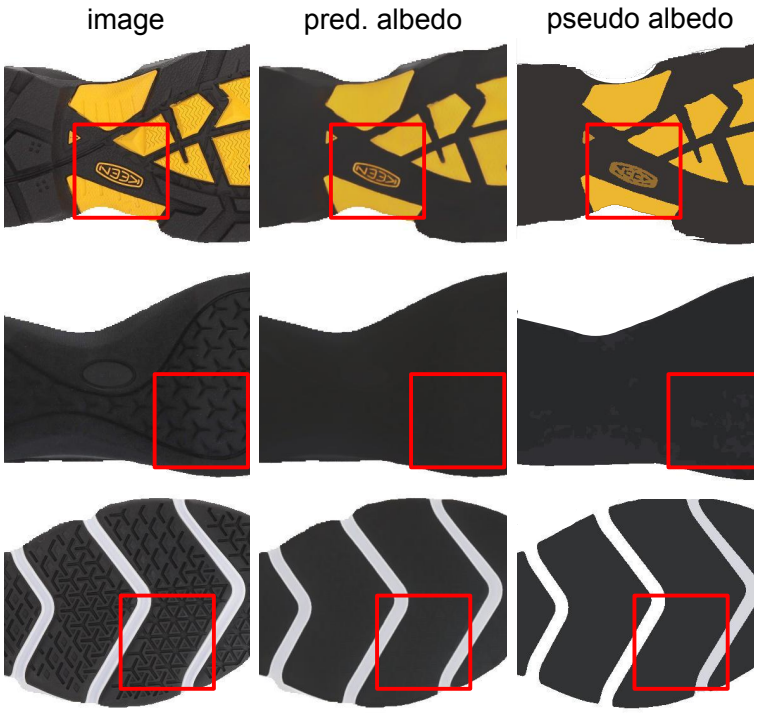

Figure A.9: Visualization of difference between predicted albedo and pseudo albedo. Given that the albedo for shoe-treads consists mostly of piece-wise constant segments, we use the mean shift clustering algorithm [31] to determine pseudo albedo. *ShoeRinsics* learns to predict albedo for real shoe-treads using the psuedo albedo as ground-truth. We do not use pseudo albedo directly instead of the albedo prediction because it is not perfect ground truth and contains deviating segment boundaries (row 1), over-segmentation (row 2), and incorrect albedo labels for segments (row 3). Our *ShoeRinsics* learns to fix these errors.

A.7 Discussion on the Pseudo Albedo

We provide pseudo supervision on the albedo prediction of real images. Fig. A.9 shows examples of pseudo albedo and the albedo predictions made by *ShoeRinsics* on real shoe-tread images. The following is a discussion on pseudo albedo generation and how pseudo albedo differs from predicted albedo.

A.7.1 Pseudo Albedo Generation

Creating pseudo albedo segments. We first group the pixels in the real image using the mean-shift algorithm [31]. To generate the pseudo albedo labels, we work with the LAB color space since it is easier to distinguish hue (A and B) from brightness (L) in this color space. Additionally, to ensure that shading does not interfere with pixel grouping, we scale the L channel by a factor of 0.15. Note that ignoring L altogether would make it difficult to distinguish between black and white. It turns out that we do not need to work on full resolution images for pixel grouping. So, we first downsize our real-shoe images to 67×150 for faster computation. After running mean-shift on the resulting shoe-tread pixels, we get an initial segmentation of the pixels in the real image. We define the color of each segment as the average color across that segment.

Refining pseudo albedo segments. The initial segmentation is very grainy as expected. Thus, we proceed to iteratively refine the segments for a maximum of 10 iterations. For each iteration we merge 'nearby' segments and update the color of the segments to reflect the segment updates. To merge segments, we find segments which are small in size and close to another segment both physically (share segment boundary) and numerically (have similar segment color). After merging segments, we update the color of the resulting segment as the average color across all the pixels in the new segment. We break the iterative refinement loop when we reach an iteration where the segmentation does not receive any updates or when the maximum iteration count (10) has been reached. Since this is a time-consuming process, we predetermine the pseudo albedo for all real shoes and save them to be used directly during training.

A.7.2 Comparing Pseudo Albedo to Predicted Albedo

It may seem counter-intuitive to learn to predict albedo when we can simply determine the corresponding ground-truth pseudo albedo. However, as we can see in Fig. A.9, pseudo albedo is only approximate and can contain deviating segment boundaries (row 1), over-segmentation (row

2), and incorrect albedo labels for segments (row 3). *ShoeRinsics* learns to fix these errors when trained using pseudo albedo as ground-truth.

Appendix B

CriSp: Leveraging Tread Depth Maps for Enhanced Crime-Scene Shoeprint Matching

B.1 Outline

Our aim is to identify shoe models resembling crime-scene impressions by comparing them to a comprehensive shoe database. Leveraging tread images from online retailers, we construct our reference database, prioritizing tread depth maps over RGB tread images for their greater relevance and informativeness [81]. As there is no dataset of crime-scene shoeprints paired with ground-truth tread depth maps, we propose learning from tread depth maps and clean shoeprints predicted from RGB tread images instead. We utilize a data augmentation module Aug to bridge the domain gap between clean and crime-scene prints, and a spatial feature masking strategy (using spatial encoder Enc and masking module M) to match shoeprint patterns with corresponding locations on tread depth maps. *CriSp* achieves significantly better retrieval results than prior methods.

In this supplementary document, we discuss the following topics:

Table B.1: Distribution of ground-truth shoe models from validation sets (val-FID and val-ShoeCase). We partition the ground-truth shoe models to assess generalization capabilities. In val-FID, there are 1152 shoe models, while val-ShoeCase comprises 16 shoe models. It's important to note that different shoe models may share tread patterns. Thus, we also distinguish between seen and unseen tread patterns during training. Val-FID encompasses 41 unique tread patterns, whereas val-ShoeCase contains 2 unique tread patterns.

	shoe models			unique tread patters		
	seen	unseen	total	seen	unseen	total
val-FID	229	923	1152	20	21	41
val-ShoeCase	2	14	16	1	1	2

- Appendix B.2 presents visualizations of retrievals by *CriSp* and also compares them with those of prior methods.
- Appendix B.3 provides a detailed quantitative comparison to state-of-the-art methods. We study generalization to unseen shoe models in Appendix B.3.1 and further detail the performance of methods on each unique shoe tread pattern in Appendix B.3.2.
- Appendix B.4 elaborates on the training process of prior methods. We investigate the performance of fine-tuning these methods using masks to simulate partial prints in Appendix B.4.1 and compare the performance of MCNCC[48] when using a reference database of shoeprints vs. tread depth maps in Appendix B.4.2.
- Appendix B.5 defines the mean average precision at K, which serves as a metric for evaluating and comparing methods.
- Appendix B.6 analyses how the ground-truth shoe models are distributed within our reference database.
- Appendix B.7 provides detailed insights into our data augmentation technique.
- Appendix B.8 shares some implementation specifics of *CriSp*.

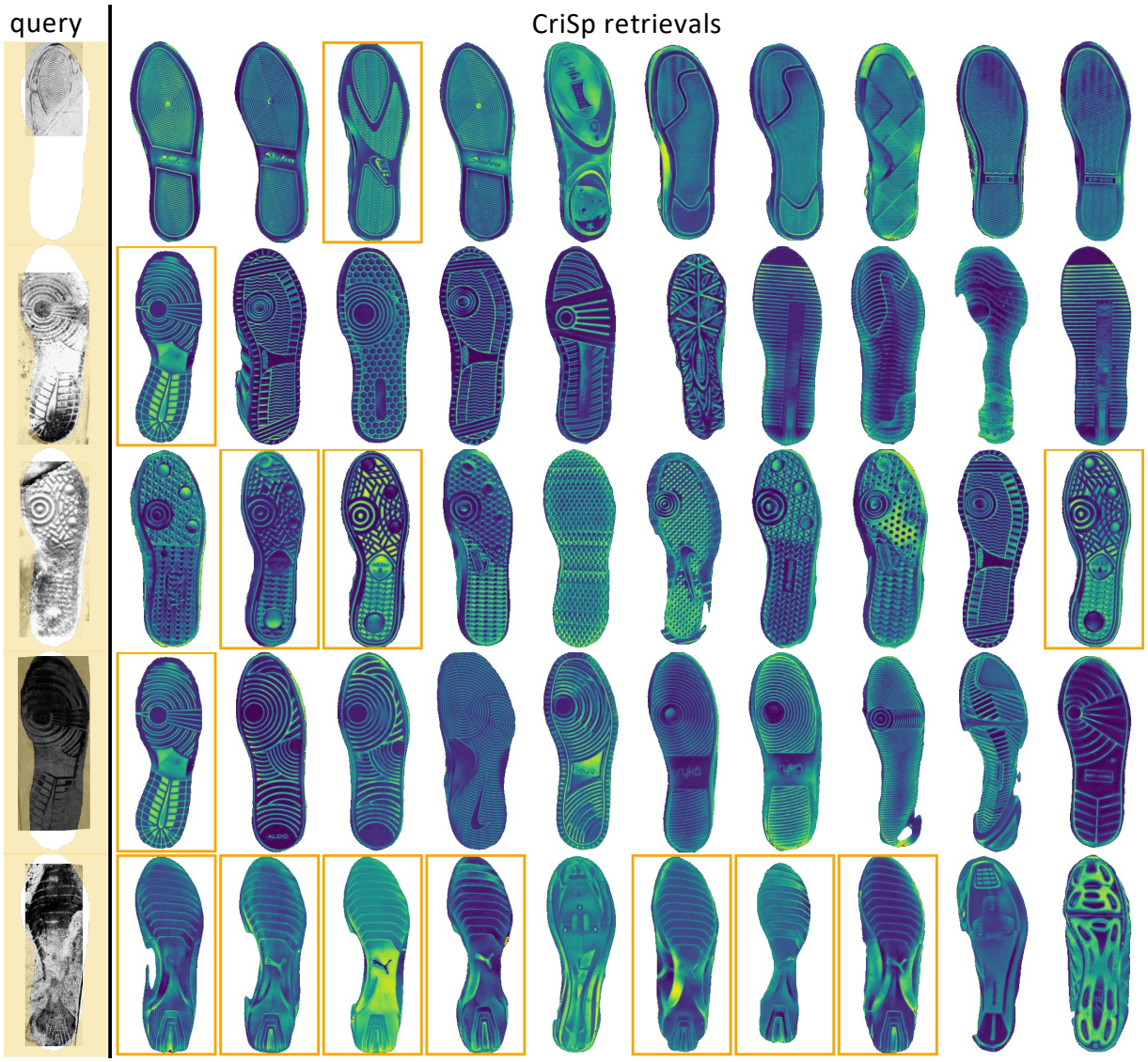

Figure B.1: Qualitative results of *CriSp* on val-FID. *CriSp* retrieves positive matches early even with partially visible or severely degraded prints.

B.2 Qualitative Results of *CriSp* and Comparison to State-of-the-art

We display visualizations in this section. Figure B.1 and B.2 show the top 10 retrievals by *CriSp* from crime-scene prints sourced from val-FID and val-ShoeCase, respectively. These illustrations demonstrate *CriSp*'s capability to retrieve positive matches even from severely degraded or

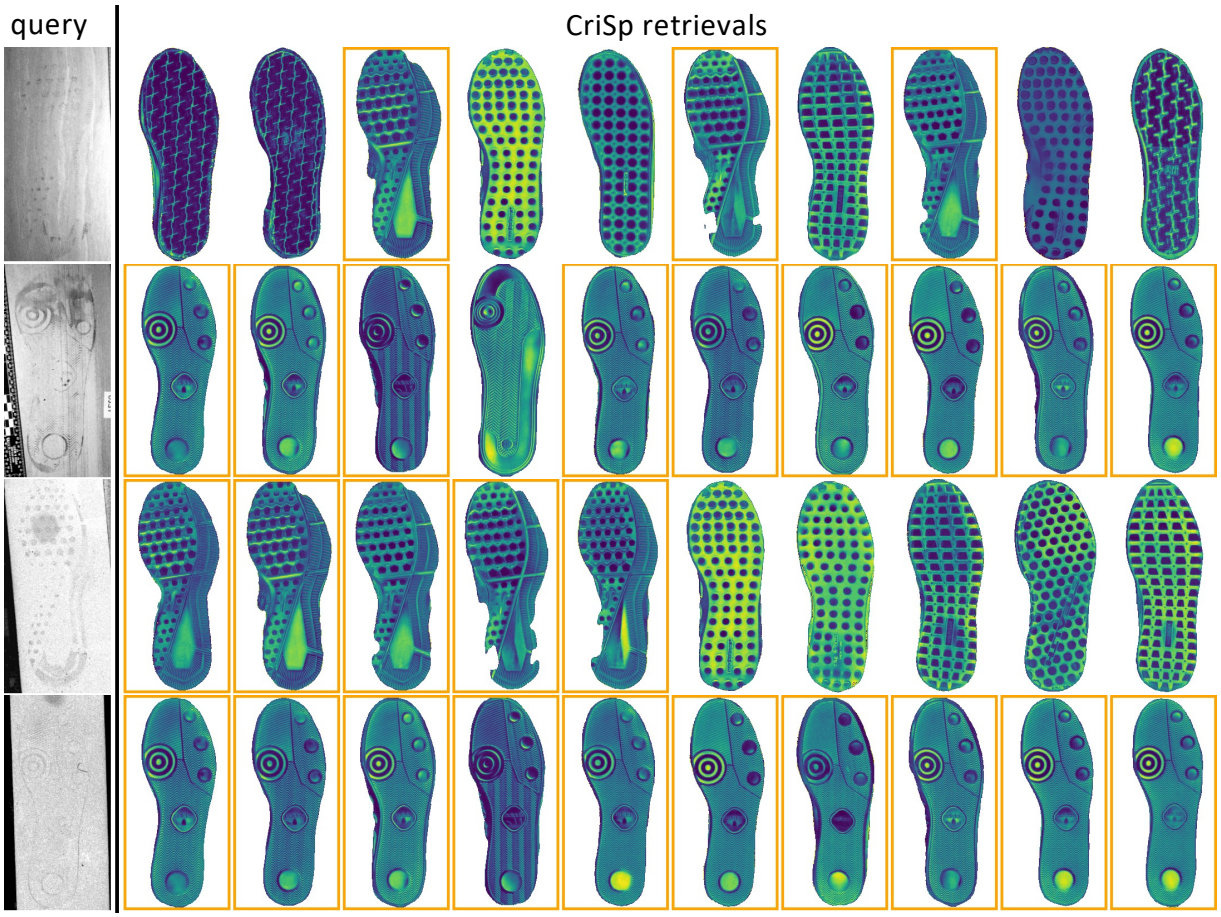

Figure B.2: Qualitative results of *CriSp* on val-ShoeCase. We show the performance on prints from two different categories: blood prints (rows 1-2) and dust prints (rows 3-4). Despite the severe degradation present in the prints, *CriSp* can retrieve positive matches early.

partially visible crime-scene shoeprints. Furthermore, Figure B.3 and B.4 provide a qualitative comparison between retrievals made by *CriSp* and those of prior methods. Notably, *CriSp* excels in matching patterns to corresponding regions on the tread, enabling it to retrieve positive matches early.

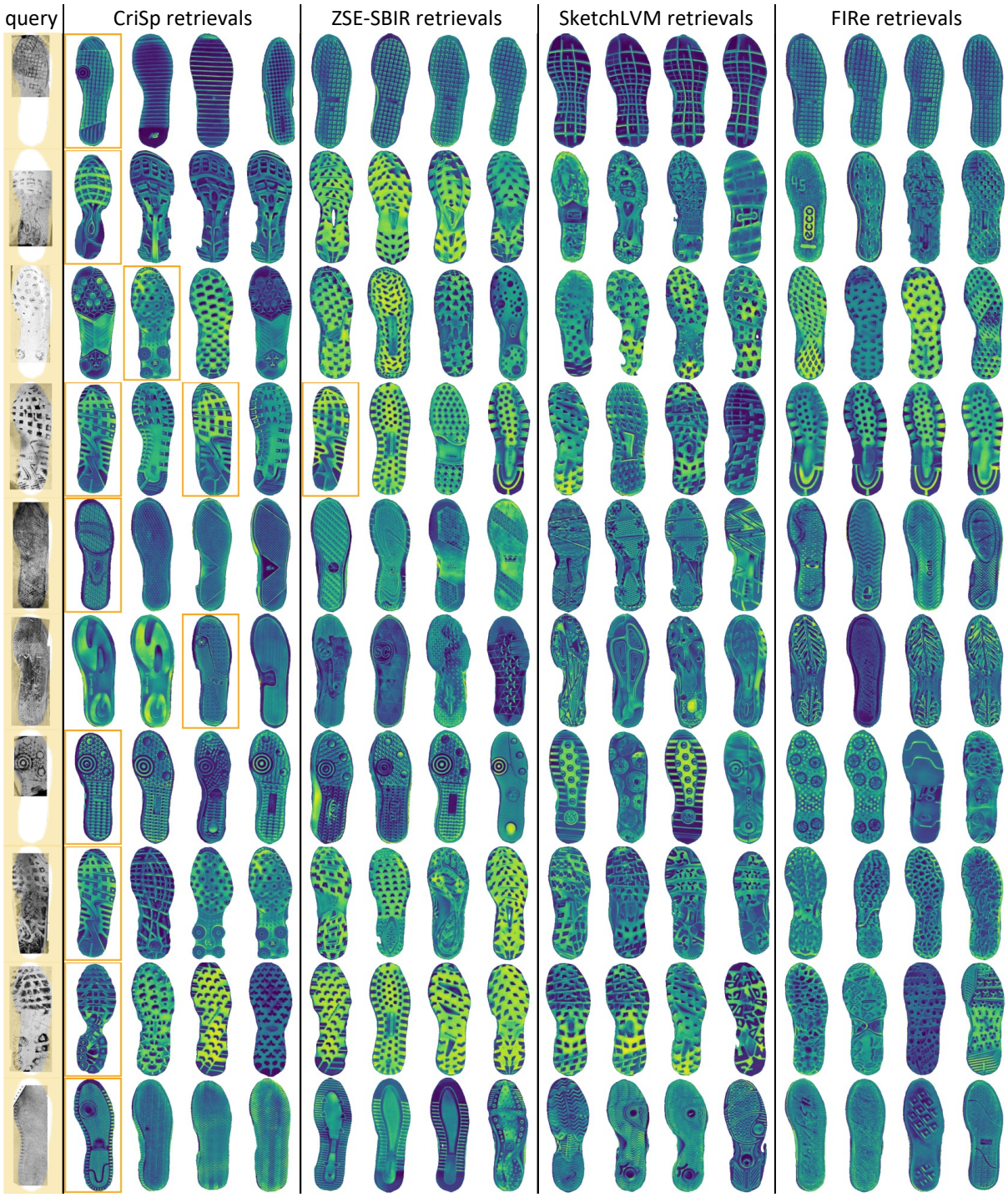

Figure B.3: Qualitative comparison to state-of-the-art on val-FID. *CriSp* outperforms prior methods by retrieving positive matches much earlier.

Figure B.4: Qualitative comparison to state-of-the-art on val-ShoeCase. *CriSp* outperforms prior methods by retrieving positive matches earlier, as evidenced by the top 6 rows displaying blood prints and the bottom 4 rows displaying dust prints.

B.3 Detailed Quantitative Comparison to State-of-the-art

B.3.1 Generalization to Unseen Shoe Models

We compare our *CriSp* to state-of-the-art methods to study generalization to unseen tread patterns.

Table B.2: Benchmarking on validation sets to study generalization. We train prior methods on our dataset with our data augmentation technique. We compare the retrieval performance of methods using mAP@100. We categorize the shoeprints in the validation sets based on whether their corresponding tread patterns were seen during training or not. Note that we perform this study in terms of seen and unseen tread patterns instead of shoe models since multiple shoe models can share the same tread pattern. Notably, *CriSp* demonstrates significantly superior performance to all prior methods on unseen tread patterns. However, ZSE-SBIR exhibits slightly better performance than *CriSp* for seen tread patterns on val-ShoeCase.

method		val-FID		val-ShoeCase	
		seen	unseen	seen	unseen
IJCV'19	MCNCC [48]	0.0002	0.0030	0.0000	0.0000
NeurIPS'20	SupCon [47]	0.0009	0.0000	0.0000	0.0000
ICLR'21	FIRe [98]	0.0671	0.0198	0.1103	0.2697
CVPR'23	SketchLVM [76]	0.0539	0.0270	0.0032	0.2770
CVPR'23	ZSE-SBIR [60]	0.1659	0.1230	**0.2653**	0.1350
	CriSp	**0.1749**	**0.2309**	0.2495	**0.4405**

since multiple shoe models can share the same tread pattern. Our findings, detailed in Table B.2, demonstrate that *CriSp* exhibits superior generalization to unseen tread patterns compared to prior methods.

B.3.2 Comparison on Unique Tread Patterns

We conduct a detailed analysis of *CriSp* relative to prior methods on each unique tread pattern from val-FID. Recall that val-FID has 41 unique tread patterns among the 1152 ground-truth shoe models. Table B.3 presents the comparison of methods based on mAP@100 for each tread pattern, where *CriSp* exhibits superior performance in the majority of cases.

Table B.3: We shoe mAP@100 for all unique tread patterns in val-FID. *CriSp* achieves the highest performance on 22 tread patterns, while ZSE-SBIR outperforms on 10 tread patterns. FIRe and SketchLVM exhibit the best performance on 1 tread pattern each.

tread pattern ID	FIRe	SketchLVM	ZSE-SBIR	CriSp
000001	0.0000	0.0000	**0.2583**	0.0027
000003	0.0000	0.0000	0.0079	**0.2333**
000004	0.3108	0.0000	0.6137	**0.6430**
000005	0.1694	0.2083	**0.5000**	0.4105
000008	0.0025	0.0010	0.0034	**0.0616**
000009	0.0000	0.0000	**0.0053**	0.0000
000010	0.0000	0.0000	0.0250	**0.5000**
000011	0.0000	0.0000	0.4275	**0.5060**
000012	0.0067	0.0022	0.0713	**0.0854**
000013	0.0348	0.1145	**0.2401**	0.0111
000016	0.0000	0.0000	0.0563	**0.0707**
000017	**0.1641**	0.0227	0.0105	0.0118
000023	0.0000	**0.3950**	0.0009	0.0148
000032	0.0000	0.0000	0.0000	0.0000
000033	0.0000	0.0000	0.2969	**0.5711**
000035	0.0000	0.0002	**0.0027**	0.0000
000045	0.0147	0.0000	0.0000	**0.2500**
000047	0.0000	0.0066	0.0000	**0.0312**
000053	0.0156	0.0000	0.0029	**0.0427**
000054	0.3148	0.0000	**0.9444**	0.0265
000055	0.0000	0.0000	0.0000	**0.3258**
000056	0.0000	0.0000	0.0000	0.0000
000062	0.0000	0.0000	0.0000	0.0000
000072	0.0018	0.0140	0.0054	**0.0821**
000074	0.0000	0.0000	0.0000	**1.0000**
000082	0.0000	0.0000	0.0000	**0.0026**
001040	0.0029	0.0000	**0.0460**	0.0044
001041	0.0000	0.0034	0.0027	**0.2000**
001044	0.2640	0.3070	0.4100	**0.5091**
001047	0.0000	0.0000	0.0000	0.0000
001048	0.0000	0.0000	**0.1036**	0.0000
001049	0.0000	0.0100	0.0238	**0.8333**
001050	0.0038	0.1704	0.0437	**0.3410**
001058	0.0000	0.0000	**0.3998**	0.0201
001062	0.0000	0.0000	0.0000	**0.4111**
001064	0.0000	0.0000	0.0000	**0.0006**
001071	0.0000	0.0000	0.0000	**0.0108**
001076	0.0000	0.0000	0.0000	0.0000
001079	0.0000	0.0000	0.0000	0.0000
001088	0.0000	0.0000	**0.5000**	0.0903
001095	0.0000	0.0000	0.0000	0.0000

B.4 Training State-of-the-art Methods

B.4.1 Fine-tuning State-of-the-art Methods Using Simulated Crime-Scene Masks

When evaluating state-of-the-art methods, we train them on our dataset and apply our data augmentation to simulate crime-scene prints during training. Here, we compare the performance of related

Table B.4: Benchmarking on real crime-scene prints from val-FID, we assess the impact of simulated partial print masks. Using hit@100 and mAP@100 as metrics, we compare the performance of prior methods trained on our dataset with our data augmentation. The mAP@100 values reveal that prior methods tend to perform better when trained without masks simulating partial prints. *CriSp* consistently achieves superior performance on both metrics, regardless of the presence of masks.

method		w/o masks		w/ masks	
		hit@100	mAP@100	hit@100	mAP@100
IJCV'19	MCNCC [48]	0.0849	0.0018	-	-
NeurIPS'20	SupCon [47]	0.0755	0.0096	0.0849	0.0001
ICLR'21	FIRe [98]	0.2075	0.0398	0.0660	0.0030
CVPR'23	SketchLVM [76]	0.1981	0.0384	0.2547	0.0445
CVPR'23	ZSE-SBIR [60]	**0.4528**	0.1412	0.4623	0.1358
	Ours	**0.4528**	**0.1765**	**0.5472**	**0.2071**

methods with and without using masks to simulate partial prints. Our findings are summarized in Table B.4, demonstrating that *CriSp* outperforms other methods in both settings.

B.4.2 Reference Database Configuration for MCNCC

When comparing MCNCC against a database of shoeprints, it yields a hit@100 of 0.0283 and mAP@K of 0.0008 on crime-scene shoeprints from val-FID. These metrics are notably lower compared to when using tread depths from the shoe database, where MCNCC achieves a hit@100 of 0.0849 and mAP@100 of 0.0018.

B.5 Evaluation Metric - Mean Average Precision at K

Mean average precision at K (mAP@K) considers both the number of positive matches and their positions in the ranking list. It rewards the system's ability to retrieve positive matches early.

MAP@K is defined as follow:

$$\text{mAP@K} = \frac{1}{Q}\sum_{q=1}^{Q} AP@K_q \tag{B.1}$$

where $AP@K_q$ is the average precision at K for query q. $AP@K$ is calculated as follows:

$$AP@K_q = \frac{1}{N}\sum_{k=1}^{K} Precision@k \times rel(k) \tag{B.2}$$

where N is the total number of positive matches for a particular query. Since we are only interested in the top K retrievals, we limit N to an upper bound of K. $Precision(k)$ is the precision calculated at each position and is defined as $\frac{pos_k}{k}$ where pos_k represents the number of positive matches in the top k retrievals. The final term, $rel(k)$, equals 1 if the item at position k is a positive match and 0 otherwise.

B.6 Distribution of Shoe Models From Validation Sets in Reference Database

Table B.1 provides insights into the distribution of ground-truth shoe models from our validation sets within the reference shoe database. Additionally, we present the count of distinct tread patterns that were either seen or unseen during training to facilitate comprehension. In Appendix B.3.1, we assess the generalization performance of our model compared to state-of-the-art methods.

B.7 Data augmentation to Simulate Crime-Scene Shoeprints

Our data augmentation module Aug simulates noisy and occluded crime-scene shoeprints from clean, fully-visible shoeprints. It introduces three types of degradation: occlusion, erasure, and noise.

- For occlusion, we simulate overlapping prints and quadrilaterals. Overlapping prints mimic the common occurrence of multiple shoeprints overlapping at a crime scene. We achieve this by randomly rotating and translating the predicted print and overlaying it onto itself. Quadrilaterals, resembling papers, rulers, or other marks, are added to simulate typical occlusions found in crime-scene shoeprint images.
- Erasure is incorporated to mimic the grainy nature of prints left at crime scenes. This involves selectively removing parts of the predicted shoeprint using either a Gaussian or Perlin distribution. Gaussian distribution is a standard choice for data augmentation, while Perlin noise provides a more nuanced representation of noise variations found in real images.
- Noise is added to represent background clutter. Gaussian or Perlin noise is overlaid on the predicted prints to simulate the clutter typically present in crime-scene images.

These degradations are applied dynamically during training, with each being optional.

B.8 Implementation Details

We use a batch size of 4, where we randomly select 4 shoe models and then include two random instances per shoe model in each batch. During our experiments, training images of size 192 x 384 are encoded to a dimension of H=6 and W = 12. We use an Adam optimizer with a learning rate of 0.1 and set $\tau = 0.07$.