Synthetic Shoe Print Generation using Pattern Mining and Texture Quilting

Thesis by
Omkar Pathak
Advised by
Charless Fowlkes

DONALD BREN SCHOOL OF INFORMATION AND COMPUTER SCIENCES
Irvine, California

ACKNOWLEDGEMENTS

The author would like to thank Charless Fowlkes, Samia Shafique and Minhaeng Lee for their support and guidance. This research project was supported by Richard Lathrop and the Information and Computer Science honors program at the University of California, Irvine.

ABSTRACT

Shoe print impressions taken at crime scenes can provide valuable intelligence for forensic analysis. Research in shoe print classification is limited due to a lack of a comprehensive dataset. This work presents a novel way to generate a synthetic dataset of shoe outsole treads using a collection of images acquired from the web. A deep Convolutional Neural Network (CNN) extracts distinctive information from small patches of shoe prints. Next, a data mining algorithm is used to find clusters of frequently occurring patterns. These patterns are synthesized into large textures that are superimposed onto shoe outsoles to generate depth maps. Several such combinations of shoe outsole depth maps are rendered to create a synthesized dataset. This work is part of a project that analyses shoe prints using a Rendered Intrinsic Network (RIN). The synthetic dataset may be open sourced for the research community.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

Shoe print images acquired at crime scenes can provide valuable intelligence just like fingerprints and DNA evidence. Recent work in forensic science technology has enabled the collection and analysis of shoe print images. Traditionally, due to a lack of automation, these raw images were manually inspected. However, several recent publications have addressed the challenges of automatically classifying shoe print impressions found at crime scenes ([1], [2], [3], [4]).

The Shoe Print Generation project is part of a project that investigates the spatial distribution and uniqueness of tread pattern features of shoe outsoles. This classification problem is especially challenging because of the variability in types of crime scene evidence (ranging from traces of dust or oil on hard surfaces to impressions made in soil) and the lack of comprehensive databases of shoe outsole tread patterns [2]. Due to such a lack of shoe outsole data, we generate a shoe print dataset that can be used for research and experimentation.

1.1 Background

To create a synthetic dataset, we use a collection of shoe print images acquired from websites of online retail stores. A pattern mining approach is deployed to find out distinct and repetitive patterns that can be observed on shoe prints. The problem of finding common objects has been crucial to image and scene classification tasks. Mid level visual elements have been extensively used to capture context information ([5], [6], [7]). In our implementation, we extract mid level visual patterns from a dataset of shoe outsole images using CNN features and data mining. The implementation of pattern mining can be found at https://github.com/omk42/Pattern-Mining. There are some other well-known pattern mining techniques like FP-growth [8], LCM [9] and KRIMP [10] algorithms. These techniques have also been applied in computer vision research ([11], [12],[13], [14], [15]). In this work, the *Apriori* algorithm is adopted. A synthetic texture is created from the repeating patterns using the *texture quilting* algorithm. This algorithm is a simple greedy approach that achieves state of the art results compared to other algorithms presented in [16], [17]. Lastly, depth map representations are generated using *K means* and several shoe print images are rendered.

Chapter 2

METHODOLOGY

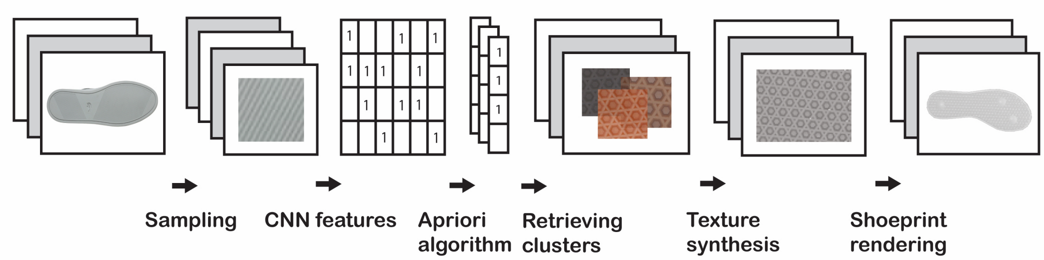

Figure 2.1: Architecture diagram.

The proposed architecture is shown in Fig. 2.1 and the various stages of implementation are described in the following sections.

2.1 Data Pre Processing

The shoe outsole data is obtained from web scraping images posted on online retail stores. The training dataset consists of 130 RGB images. The first step in preprocessing the data is sampling the images into 48×48 patches with a step size of 24. Only masked patches containing the shoe print are used for training. As per the requirements of the Berkeley Vision and Learning Center (BVLC) Reference CaffeNet model, the input is processed in the following order [18].

- The dataset means of red, green and blue channels are subtracted from the channels of the input image. This ensures that the mean of feature pixels is 0.
- The input image is transposed to channels $\times$ height $\times$ width.
- The color channels are swapped from RGB to BGR.
- The input image is reshaped to 227×227 to match the input dimensions of the Caffe model.

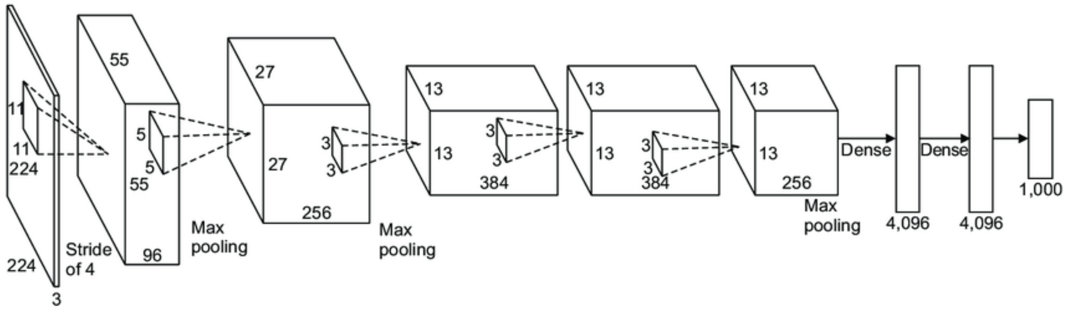

Figure 2.2: BVLC Reference CaffeNet architecture [22].

2.2 Convolutional Neural Network (CNN) Features

State of art results have been achieved by using feature encoding methods on activations from fully connected layers of deep CNNs ([19], [20]). It is demonstrated that discriminative information from image patches can be extracted from their CNN activations [21].

The pretrained BVLC Reference CaffeNet model is used for feature extraction [18]. This Caffe model is a modification of the AlexNet [23]. Unlike AlexNet, the BVLC Reference CaffeNet is trained without relighting and the pooling and normalization layers are interchanged. Fig. 2.2 shows the architecture of the CaffeNet model. The 4096 non-negative weights from the second fully-connected layer (fc7) are saved for each input patch.

2.3 Apriori algorithm for Mining Association Rules

The apriori algorithm was initially designed for market basket analysis. It uses transactions that are subsets (s) of a global collection of N items (D). i.e. ($s \subset D$). It is used for mining frequent itemsets and association rules over a database of transactions [24]. The algorithm uses a support threshold ($\mathrm{supp}_{\mathrm{min}}$) to identify patterns (P) that are subsets of at least $\mathrm{supp}_{\mathrm{min}}$ transactions (T) from the database (Eq. 2.1). For association rule mining, the algorithm uses a confidence threshold ($\mathrm{conf}_{\mathrm{min}}$) that ensures that at least $\mathrm{conf}_{\mathrm{min}}$ patterns contain a certain item (a) (Eq. 2.2) [21].

$$supp(P) = \frac{\{T | T \in D, P \subseteq T\}}{N} \tag{2.1}$$

$$conf(P \rightarrow a) = \frac{supp(P \cup \{a\})}{supp(P)} \tag{2.2}$$

Example: Consider the case where there are 4 items in the set (i.e., A = {1, 2, 3, 4 }) and 5 transactions in the database (D),

- T1 = {3, 4},
- T2 = {1, 2, 4},
- T3 = {1, 4},
- T4 = {1, 3, 4},
- T5 = {1, 2, 3, 4},

The value of supp({1, 2, 4}) is 0.4 as the pattern{1, 2, 4} appears in 2 out of 5 transactions (i.e., {T2, T5 }). The confidence value of the rule {1, 2, 4} → 3 is 0.5 (i.e., conf({1, 2, 4} → 3) = 0.5) as 50% of the transactions containing {1, 2, 4} also contains the item 3 (i.e., {T5}).

2.4 Shoe Print Patches Pattern Mining

Section 2.2 describes the process of extracting 4096 dimensional features from shoe print image patches. On the other hand, the previous section 2.3 goes over the apriori algorithm and its requirements. A database of itemsets or transactions is required to find frequently occurring shoe print patterns. To represent the CNN activations of image patches as transactions, the indices of the K largest features are used. In addition to the K largest elements, another number (4097) is added to the transactions which is later used for finding association rules that contain 4097. As an example, let's consider an image patch whose CNN activations are [0,3,0,12,10]. If K is 2, the largest elements (12 and 10) are at positions 3 and 4. Hence, the transaction corresponding to that image patch is {3,4,4097}.

It has been shown in [21] that binarization of CNN activations does not lose the discriminative property when K is small. Hence, similar to the implementation in [21], the value of K is set to 20.

$$supp(P \cup 4097) = supp(P) \times conf(P \rightarrow 4097) > supp_{min} \times conf_{min} \quad (2.3)$$

Mid level visual patterns can facilitate the creation of synthetic datasets only if the extracted patterns are distinct and representative. Distinct patterns mean that the

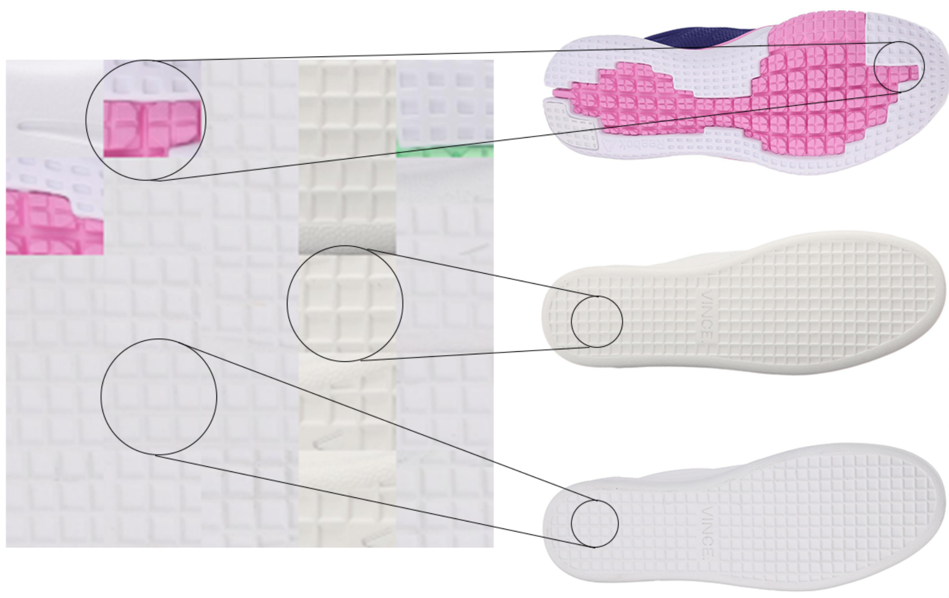

Figure 2.3: Mosaic with its corresponding shoe print images.

clusters of patches should be visibly distinguishable from each other. On the other hand, representativeness requires that the identified patterns occur repeatedly in the training dataset. The apriori algorithm is chosen as it provides a direct way to satisfy both these requirements. As seen in eq 2.3, the support and confidence thresholds can be fine tuned to maintain the frequency and uniqueness of the shoe print patterns [21].

2.5 Retrieving Clusters

The transaction database from section 2.4 is used by the apriori algorithm to generate association rules. The algorithm outputs rules that contain subsets of feature indices that frequently occur among shoe print patches. All patches whose K largest element indices include these indices form a cluster.

Example: Consider that {1,4} is a rule (R1) and the transaction database is as follows-

- T1 = {1,2,3,4},
- T2 = {1,2,3,5},

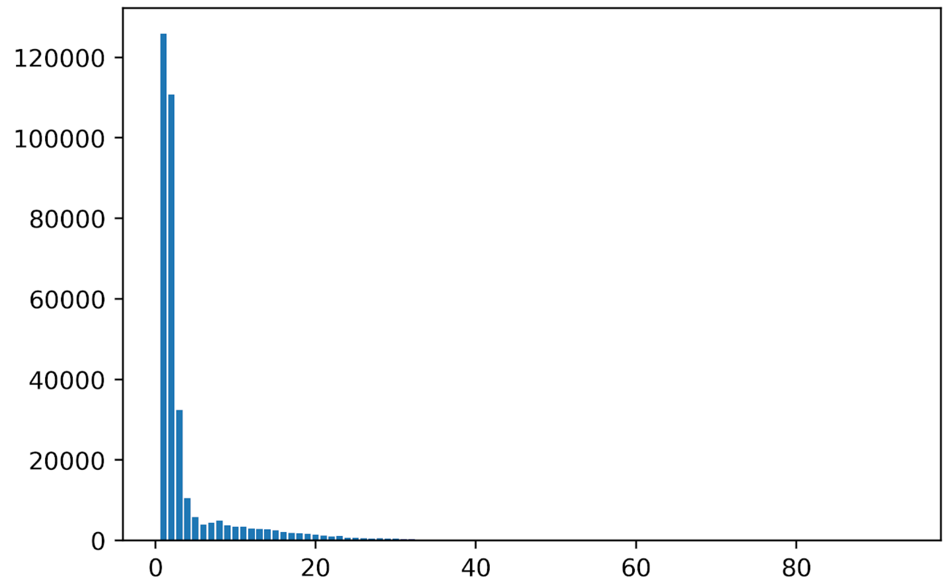

Figure 2.4: Cluster statistics.

- T3 = {1,3,4,5},
- T4 = {1,2,5,6},
- T5 = {1,4,5,6},

The rule R1 is a subset of transactions T1, T3 and T5. Hence, the cluster (C1) corresponding to R1 will contain patches 1, 3 and 5. All rules that are subsets of other rules are removed in order to avoid duplicate clusters. For example, if {1} is a rule (R2) its cluster (C2) will contain patches 1, 2, 3, 4 and 5. As can be seen, clusters C1 and C2 contain duplicate patches. Hence, R2 and C2 are disregarded.

Visualization- For easy visualization, patches belonging to a cluster are arranged in a 5×5 mosaic. Fig 2.3 illustrates a mosaic (on the left) whose 3 patches are traced back to their parent shoe prints (on the right).

Table 2.1 shows the cluster statistics obtained using several sets of minimum confidence, minimum support and length of features parameters. The changes in average patches and images per cluster are proportional to the minimum support parameter from the apriori algorithm. This happens due to the fact that large support parameters force the algorithm to find more repetitive patches. By lowering the length of

Minimum item set support	**0.1**	**0.3**	**0.8**
Minimum feature length	**13**	**11**	**8**
Maximum feature length	**15**	**13**	**10**
Average patches per cluster	20.34	55	182
Number of clusters	2154732	338675	55919
Average images per cluster	1.60	3.82	29.57

Table 2.1: Cluster statistics for various apriori parameters.

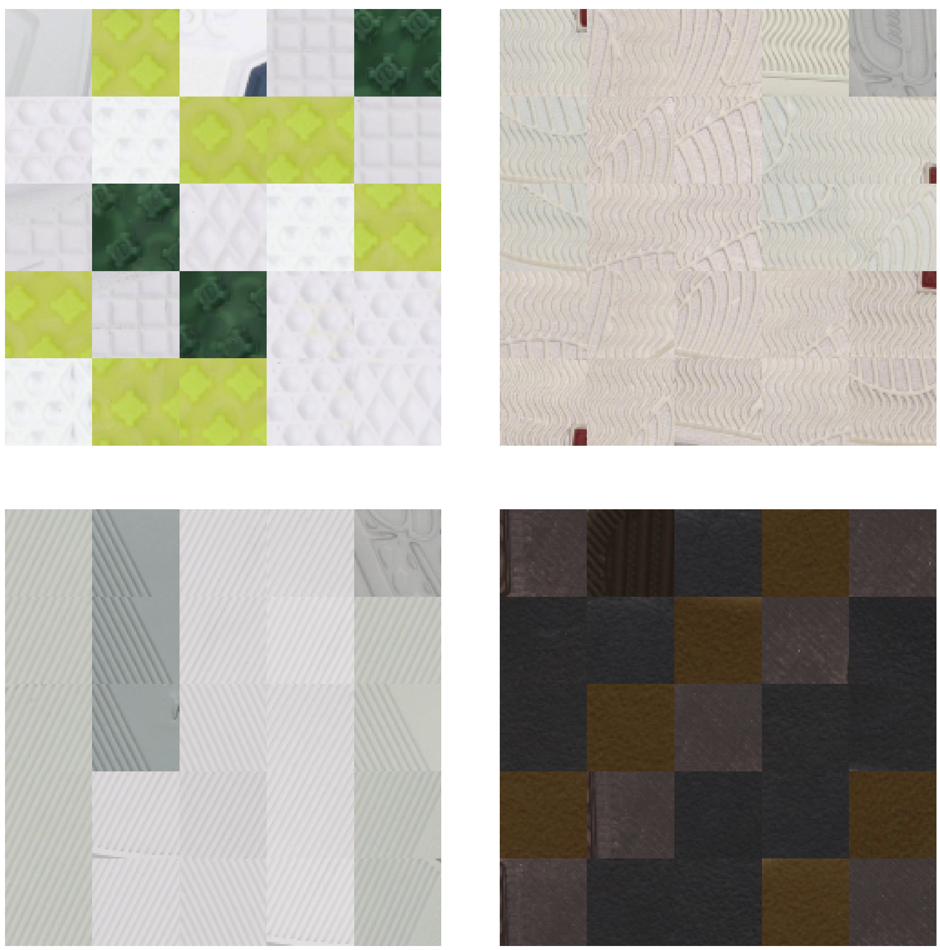

Figure 2.5: Sample mosaics.

the repeating itemsets the algorithm is able to compensate for greater coverage. We choose to apply the parameters as mentioned in table 2.2.

Min item set support	Min confidence	Min feature length	Max feature length
0.3 %	50 %	11	13

Table 2.2: Apriori parameters.

Our method was able to retrieve 338675 clusters of patches with an average of 55 patches in every cluster. Fig 2.4 shows the distribution of the number of unique shoes in clusters. The x axis is the number of unique shoes in a cluster and the y axis contains the count of clusters. On average, every cluster contains patches from 3.82 out of 130 images from the dataset. Fig 2.5 displays several mosaics retrieved by the algorithm.

2.6 Image Quilting algorithm

The authors in [25] have presented a simple greedy approach to synthesize a big image by stitching together small image patches. Given a collection of similar texture patches, the algorithm produces a bigger sized texture image as seen in Fig 2.6. The patches of images are pasted greedily next to each other with some overlap. At every stage, the next patch is selected based on the similarity of the overlapping region. To provide a smooth transition between patches, the overlapping portion is raggedly cut. The authors proposed a dynamic programming approach to determine the minimum error boundary cut in the overlapping region. The steps of the algorithm are as follows-

- Start with the top left patch in the output image and continue processing in a raster order.
- At every stage, randomly pick a patch from the database that satisfies the error tolerance of similarity in the overlapping region.
- Compute the minimum error boundary cut and paste the new patch along the cut. Repeat till the end.

2.7 Shoe Print Texture Synthesis

Section 2.5 describes the process of mining clusters of similar patches. In this section, we apply the image quilting algorithm to synthesize a shoe print texture from the retrieved clusters of patches. The mosaics shown in Fig 2.5 simply paste random

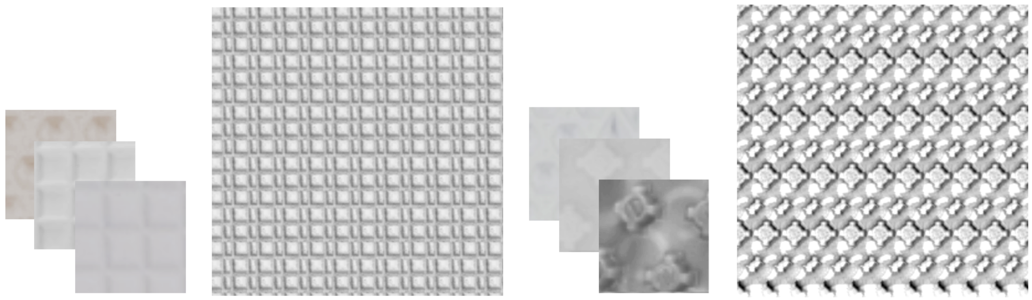

Figure 2.6: Synthesized texture using image patches.

patches from clusters together. However, the texture needs to be homogeneous to superimpose it on a shoe outsole. Hence, the image quilting algorithm is used. Fig 2.6 shows the results of image quilting algorithm on several clusters of shoe print patches. The overlap length is experimentally determined as 20.

2.8 Depth Generation and Rendering

This section generates a depth map from the shoe print texture image and renders it using Mitsuba [26]. In [27], K means method and connected components are used for assigning depths in spatial and temporal domains. Similarly, this implementation applies K-means on brightness values to assign random depths to every pixel. Fig 2.7 shows depth maps of shoe print textures in Meshlab [28]. As can be seen, three hemispheres, a boundary and some blurriness is added to the shoe prints to account for variations seen in forensic data. A few depth maps are rendered in Mitsuba and their outputs are shown in Fig 2.8.

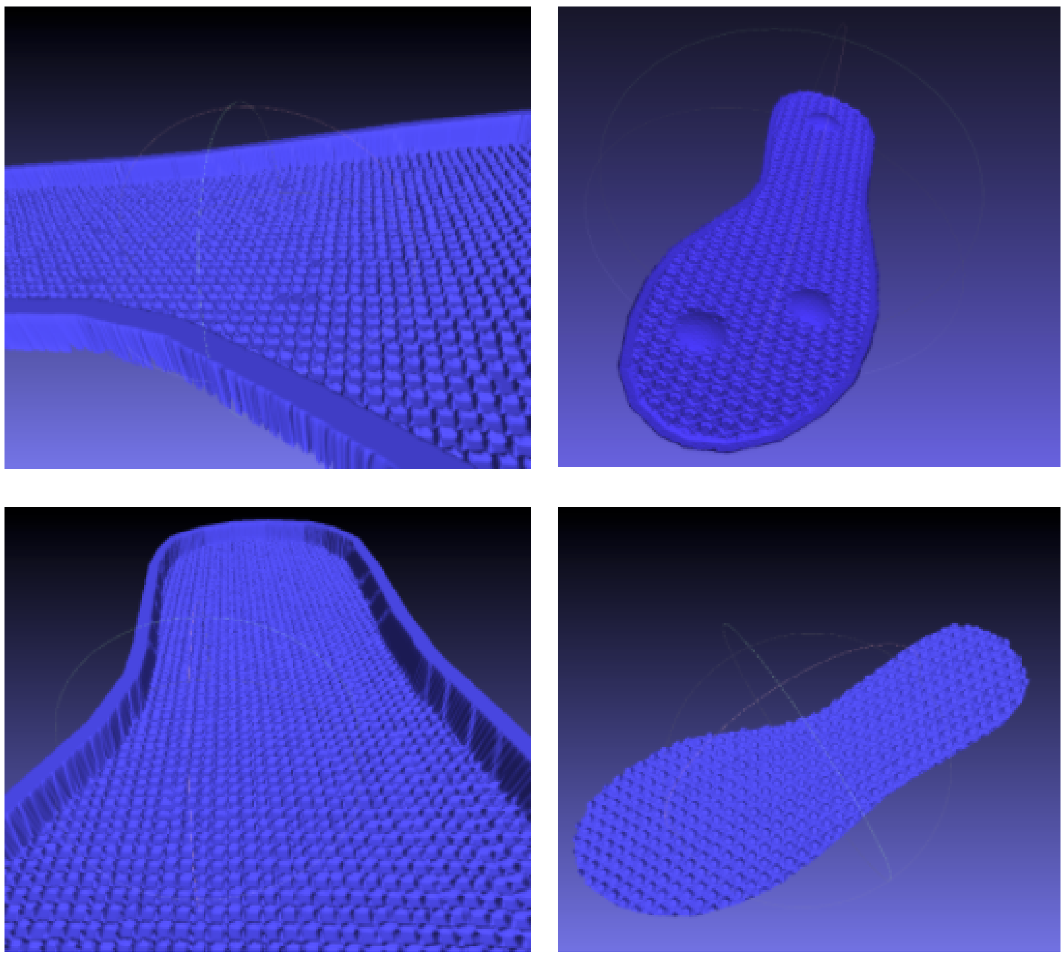

Figure 2.7: Depth maps of synthesized shoe print textures.

Figure 2.8: Rendered shoe print images in grayscale.

Chapter 3

CONCLUSION

The goal of this project is to create a pipeline for synthesizing realistic shoe print images that can be used for classification research and experiments. In this implementation, the CaffeRef CNN model is used for feature extraction. However, several other ways of feature extraction such as Histogram of Oriented Gradients (HOG), auto encoders, statistical methods and other CNN models such as the VGG-VD can also be used. This paper shows the experimental results of the apriori algorithm with several parameters. However, one noteworthy drawback of this algorithm is that its performance degrades as the length of the desired frequent itemset increases [21]. Hence, other data mining methods (section 2.3) may also be applied to get frequent item sets at scale. The texture quilting algorithm is a simple solution that produces visually accurate synthetic images. However, more complex solutions described in section 2.6 can make this process more robust.

In future, we plan to use our synthesized dataset for analysis using a Rendered Intrinsic Network (RIN) [29]. The dataset may also be open sourced to the research community.

BIBLIOGRAPHY

[1] G. AlGarni and M. Hamiane, "A novel technique for automatic shoeprint image retrieval," *Forensic Science International*, vol. 181, no. 1–3, pp. 10–14, Oct. 2008. DOI: 10.1016/j.forsciint.2008.07.004.

[2] B. Kong, J. Supancic, D. Ramanan, and C. C. Fowlkes, "Cross-domain image matching with deep feature maps," *International Journal of Computer Vision*, vol. 127, no. 11–12, pp. 1738–1750, Jan. 2019. DOI: 10.1007/s11263-018-01143-3. [Online]. Available: https://arxiv.org/pdf/1804.02367.pdf.

[3] Alexander, Bouridane, and Crookes, *Automatic classification and recognition of shoeprints*, Jul. 1999. DOI: 10.1049/cp:19990401. [Online]. Available: https://ieeexplore.ieee.org/document/791138?arnumber=791138.

[4] P. de Chazal, J. Flynn, and R. Reilly, "Automated processing of shoeprint images based on the fourier transform for use in forensic science," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 341–350, Mar. 2005. DOI: 10.1109/TPAMI.2005.48. [Online]. Available: https://ieeexplore.ieee.org/document/1388261?arnumber=1388261.

[5] C. Doersch, A. Gupta, and A. Efros, *Mid-level Visual Element Discovery as Discriminative Mode Seeking*. [Online]. Available: http://graphics.cs.cmu.edu/projects/discriminativeModeSeeking/nips2013.pdf.

[6] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, *Blocks that Shout: Distinctive Parts for Scene Classification*. [Online]. Available: https://www.robots.ox.ac.uk/~vedaldi/assets/pubs/juneja13blocks.pdf.

[7] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," *arXiv:1205.3137 [cs]*, Aug. 2012. [Online]. Available: https://arxiv.org/abs/1205.3137.

[8] G. Grahne and J. Zhu, "Fast algorithms for frequent itemset mining using fp-trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 10, pp. 1347–1362, Oct. 2005. DOI: 10.1109/tkde.2005.166.

[9] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "An efficient algorithm for enumerating closed patterns in transaction databases," *Discovery Science*, pp. 16–31, 2004. DOI: 10.1007/978-3-540-30214-8_2.

[10] J. Vreeken, M. van Leeuwen, and A. Siebes, "Krimp: Mining itemsets that compress," *Data Mining and Knowledge Discovery*, vol. 23, no. 1, pp. 169–214, Oct. 2010. DOI: 10.1007/s10618-010-0202-x.

[11] B. Fernando, E. Fromont, and T. Tuytelaars, "Mining mid-level features for image classification," *International Journal of Computer Vision*, vol. 108, no. 3, pp. 186–203, Feb. 2014. DOI: 10.1007/s11263-014-0700-1. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00968299/file/ijcv2014.pdf.

[12] B. Fernando, E. Fromont, and T. Tuytelaars, *Effective Use of Frequent Itemset Mining for Image Classification*. [Online]. Available: https://homes.esat.kuleuven.be/~konijn/publications/2012/ECCV_ID10.pdf.

[13] B. Fernando and T. Tuytelaars, *Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation*, 2013. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2013/html/Fernando_Mining_Multiple_Queries_2013_ICCV_paper.html.

[14] K. Rematas, B. Fernando, F. Dellaert, and T. Tuytelaars, *Dataset fingerprints: Exploring image collections through data mining*, 2015. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2015/html/Rematas_Dataset_Fingerprints_Exploring_2015_CVPR_paper.html.

[15] J. Yuan, Y. Wu, and M. Yang, *Discovery of Collocation Patterns: from Visual Words to Visual Phrases*. [Online]. Available: http://users.eecs.northwestern.edu/~yingwu/papers/conference/2007/CVPR07d_final.pdf.

[16] B. Guo, H. Shum, and Y.-Q. Xu, "Chaos mosaic: Fast and memory efficient texture synthesis," *www.microsoft.com*, Apr. 2000. [Online]. Available: https://www.microsoft.com/en-us/research/publication/chaos-mosaic-fast-and-memory-efficient-texture-synthesis/.

[17] E. Praun, A. Finkelstein, and H. Hoppe, "Lapped textures," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000. DOI: 10.1145/344779.344987.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[19] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, *Deep filter banks for texture recognition, description, and segmentation*, 2015. [Online]. Available: https://arxiv.org/abs/1507.02620.

[20] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," *arXiv:1403.1840 [cs]*, Sep. 2014. [Online]. Available: https://arxiv.org/abs/1403.1840.

[21] Y. Li, L. Liu, C. Shen, and A. v. d. Hengel, "Mining mid-level visual patterns with deep cnn activations," *arXiv:1506.06343 [cs]*, May 2016. [Online]. Available: https://arxiv.org/abs/1506.06343.

[22] V. Albu, "Measuring customer behavior with deep convolutional neural networks," 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*. 2012. [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[24] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules*. [Online]. Available: http://www.vldb.org/conf/1994/P487.PDF.

[25] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, 2001. DOI: 10.1145/383259.383296.

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

[27] Y.-L. Chang, C.-Y. Fang, L.-F. Ding, S.-Y. Chen, and L.-G. Chen, *DEPTH MAP GENERATION FOR 2D-TO-3D CONVERSION BY SHORT-TERM MOTION ASSISTED COLOR SEGMENTATION*. [Online]. Available: https://projet.liris.cnrs.fr/imagine/pub/proceedings/ICME-2007/pdfs/0001958.pdf.

[28] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds., The Eurographics Association, 2008, ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[29] M. Janner, J. Wu, T. Kulkarni Deepmind, and J. Tenenbaum, *Self-Supervised Intrinsic Image Decomposition*. [Online]. Available: http://people.csail.mit.edu/janner/papers/intrinsic_nips_2017.pdf.