Learning a Similarity Measure for Striated Toolmarks using Convolutional Neural Networks

Manuel Keglevic and Robert Sablatnig

Computer Vision Lab - TU Wien, Favoritenstr. 9/183-2, A-1040 Vienna, Austria
mkeglevic@caa.tuwien.ac.at

Keywords: forensics, toolmarks, deep learning, neural networks

Abstract

We propose TripNet a method for calculating similarities between striated toolmarks. The objective for this system is to distinguish the individual characteristics of tools while being invariant to class and sub-class characteristics, and varying parameters like angle of attack. Instead of designing a handcrafted feature extractor customized for this task we propose the use of a Convolutional Neural Network (CNN). With the proposed system 1D profiles extracted from images of striated toolmarks are mapped into an embedding. The system is trained by minimizing a triplet loss function so that a similarity measure is defined by the L_2 distance in this embedding. The performance is evaluated on the NFI Toolmark database containing 300 striated toolmarks of screwdrivers published by the National Forensic Institute of the Netherlands. The proposed system is able to adapt to a large range of angles of attack between 15°and 75°, achieving a Mean Average Precision (MAP) of 0.95 for toolmark comparisons with differences in angle of attack of 15°to 45°; for differences of 15°to 60°a MAP of 0.78 is achieved.

1 Introduction

Since the validity of comparative forensic examination of toolmarks has been challenged in court, papers have been published with focus on obtaining statistical support for the notion of the *uniqueness* of toolmark patterns [1], i.e. the existence of "measurable feature with high degree of individuality" [2]. Even though the requirement for such *uniqueness* is debatable [3], this led to a variety of methodologies [2, 4, 5, 6, 7, 8, 9] for automatically, and objectively [4], comparing striated toolmarks.

The input for these algorithms are 1D profiles extracted from either 2D images or 3D surface scans of the striated toolmarks. In Figure 1 two images with superimposed profiles from the same tool at different angles of attack are depicted. After preprocessing, similarity scores are commonly computed using the Cross-Correlation (CC) as proposed by the National Institute of Standards and Technology (NIST) for comparing ballistic toolmarks [10, 11]. This approach is either applied globally on the whole profile [4, 5, 6] or locally [8]. Another similarity measure based on locally normalized squared distances, the so-called *relative distance*, is propose by Bachrach et al. [2]. These similarity measures are then used to predict if a given pair of toolmarks are matches or non-matches, i.e. made from the same tool or not, respectively.

In contrast to computing a similarity measure, Petraco et al. [9] propose a classification approach based on machine learning. In a first step, Principle Component Analysis (PCA) and Linear discriminant analysis (LDA) are used for dimensionality reduction of the input profiles. The identity of the tool, i.e. the class, is then predicted using Support Vector Machines (SVM).

The common challenge for comparing striated toolmarks lies in distinguishing individual characteristics of the tool while being invariant to class and sub-class characteristics [4]. Further, parameters like angle of attack (from now on referred to as α), substrate material and axial rotation have a major impact on toolmarks [5]. Baiker et al. [4] showed that when comparing toolmarks with different α, for differences of 30°the error rate is more than a magnitude higher than for differences of 15°, i.e. the false discovery rate increases from 3.00% to 36.67%.

We propose the use of Convolutional Neural Networks (CNN) [12]. Instead of designing a handcrafted feature extractor which on the one hand is sensitive enough to distinguish the fine grained individual characteristics and on the other hand is robust enough to be invariant to changes in the aforementioned parameters, it is trained from end to end. Recent advances in deep learning have shown that CNNs outperform the state-of-the-art of handcrafted feature descriptors [13]. For comparing images Chopra et al. [14] propose a siamese architecture where two identical networks with shared weights are used to learn a low dimensional representation of images. In this feature space (*embedding*) similarities between face images are computed using the L1 norm. Further, a similar architecture is

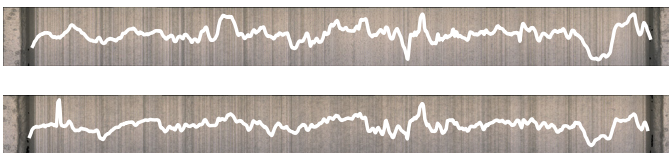

Figure 1: Superimposed 1D profiles extracted from 3D surface scans onto 2D images of NFI Toolmarks; (top) 15 degree, (bottom) 30 degree.

applied by Zagoruzkoa and Komadaski [15] for matching local image patches. Schroff et. al [16] propose the use of a triplet loss function minimizing the distance between an anchor and a positive (match) while maximizing the distance between the anchor an a negative sample (non-match). Similar architectures are likewise used to distinguish faces [17] and local image patches [18].

In this paper we propose the use of a CNN called "TripNet" for matching toolmarks profiles extracted from 2D images. To allow a fast computation of similarities the triplet loss function described by Balntas et al. [18] is applied. This way, instead of computing the distances between all possible toolmark pairs using the CNN (NxN comparisons) each toolmark is mapped into the embedding where the L_2 distance corresponds to a similarity score. Our evaluation is based on the NFI database of 300 striated screwdriver toolmarks published by Baiker el al. [4].

This paper is divided into the following sections: Firstly in Section 2, a curvature matching method is proposed as our baseline approach. Secondly in Section 3, the loss function, the network architecture and the design decisions behind are described. Thirdly in Section 4, our baseline is evaluated against the results by Baiker et al. [4] and the performance improvements of TripNet are shown. We conclude with the advantages and disadvantages of our approach and future work is discussed.

2 Baseline

Our baseline approach is based on the elastic shape metric proposed by Srivastava et al. [19]. For comparing shapes of closed and open curves in $\mathbb{R}^n$ the distance is defined as a combination of bending and stretching deformations. In contrast to other elastic shape metrics, the curve is represented by the square-root-velocity (SRV) function to reduce it to an L^2 metric. All curves are scaled to unit length in order to achieve scale invariance. These open curves with unit lengths are then represented by points on a unit hypersphere in this *pre-shape-space* $L^2(D, \mathbb{R}^n)$. The distance between two curves is then defined by the length of the minimizing geodesic between their point representations in *pre-shape-space*. Since this *pre-shape-space* is not invariant to rotation and re-parameterization an additional optimization step is performed afterwards to compute the distances in *shape-space*.

This approach is directly applied to the NFI Toolmark profiles. The only preprocessing step performed is downsampling to 800 points which corresponds to the minimal wavelength used by Baiker et al. [4]. In Figure 2 the result of this approach is visualized on an example using the same profiles as depicted in Figure 1. The corresponding points are shown as connections between the curves.

3 TripNet

Our proposed neural network TripNet is based on the work of Balntas et al. [18]. The architecture is depicted in Figure 3. Similar to siamese networks [14], there are multiple branches

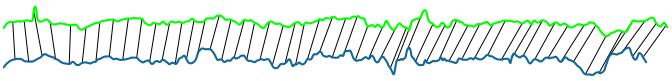

Figure 2: Result of applying the elastic shape matching operation to the profiles of tool 1A for α of 15°(bottom) and 30°(top).

with shared weights. The training is performed by forwarding three input samples (a triplet $T = \{x_{p_1}, x_{p_2}, x_n\}$) through these branches, i.e. they are mapped into the embedding $f(x_i)$. Two samples are chosen from one class and another one from a different class, i.e. x_{p_1}, x_{p_2} and x_n, respectively. The results are then combined in the loss function and the error is back-propagated.

The dimension of this embedding $f(x)$ can be controlled by changing the size of the last layer in the branches. Since, the weights are shared only one branch is needed after the training. As the loss function minimizes the euclidean distance between matching samples, the L^2 norm can be used to measure distances in the embedding. Therefore, efficient algorithms for calculating L_2 distances can be applied [18]. Additionally, the storage requirements are directly controlled by changing the dimension of the embedding.

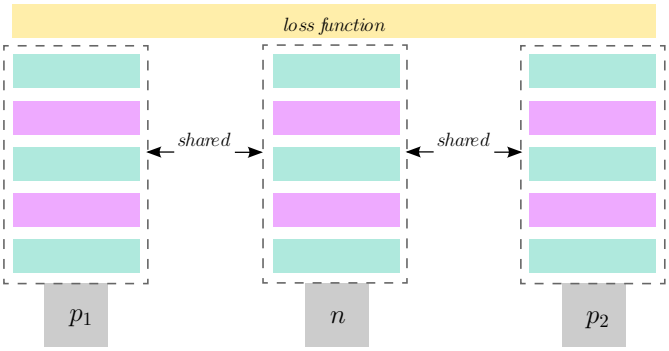

Figure 3: Triplet architecture

3.1 Triplet Loss

In contrast to other triplet loss functions like the SoftMax Ratio proposed by Hoffer et al. [20] which only takes one negative distance into account, all three distances between the samples are used in [18]:

$$\begin{aligned} \Delta^+ &= \|f(x_{p_1}) - f(x_{p_2})\|_2 \\ \Delta_1^- &= \|f(x_{p_1}) - f(x_n)\|_2 \\ \Delta_2^- &= \|f(x_{p_2}) - f(x_n)\|_2 \end{aligned} \tag{1}$$

with the triplet $T = \{x_{p_1}, x_{p_2}, x_n\}$ and the embedding $f(x)$. Instead of forcing the distance Δ^+ to be just smaller than Δ_1^-, it is forced to be smaller than $\Delta^* = min(\Delta_1^-, \Delta_2^-)$. The difference is illustrated in Figure 4.

The loss is then defined as [18]

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

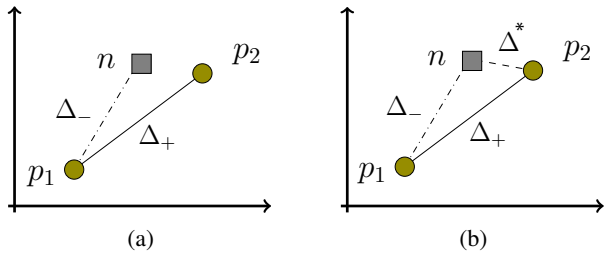

Figure 4: SoftMax Ratio (a) compared to SoftPN (b) [18].

which is implemented using a Softmax layer and the Mean Square Criterion. The selection of training samples is simplified by this approach since soft negative mining is performed implicitly [18].

3.2 CNN Architecture

The architecture of the CNN is depicted in Table 1. As our input images are 1x800 the convolutional and pooling layers have one dimensional input regions.

Each convolutional layer is followed by batch normalization to decrease the dependency on input normalization and initialization of the network [21]. The size of the convolutions and the number of feature maps as well as the size of the pooling layers were empirically evaluated. The best results are achieved with 1x5 convolutions and 1x3 pooling with 64 feature maps in the first convolution and 32 in the second. In contrast to [18] Rectified Linear Units (ReLU) [22] and average pooling [12] are used since this setup performs better. However, for the last layer a smooth output is ensured by a Tanh activation function. To additionally fight overfitting due to our small dataset a Dropout [23] layer is added at the end with a probability of 0.5.

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

Table 1: Architecture of the CNN branches.

3.3 Implementation Details

The training and evaluation of TripNet is implemented in Torch[1]. Similarly to the processed 1D profiles used for the baseline the 2D images are downscaled to a height of 800 pixels. For training random vertical 1x800 crops are taken to increase the variability of the samples. The triplet creation is done on-line, i.e. not created beforehand but during training. Triplets with positive samples extracted from the same image are discarded to remove trivial samples. Min/max-normalization and mean pixel subtraction is performed as a preprocessing step. For evaluation only center crops are used for reproducibility. Similarity scores are computed by computing the L_2 distance between toolmarks of the testset and the trainingset in the embedding.

The optimization is done using Stochastic Gradient Decent with a learning rate of 0.01, weight decay of 10^{-4} and momentum of 0.9.

[1] https://github.com/torch/torch7

4 Evaluation

In contrast to other works on toolmarks [4, 2] this paper approaches the evaluation in terms of Information Retrieval (IR). In IR a user expresses an *information need* using a set of queries and retrieves *relevant* and *nonrelevant* documents out of a *document collection* [24]. In case of toolmarks the *information need* can be expressed as the search for a similar toolmark, i.e. the search for a toolmark made by the same tool. This way *relevant* and *nonrelevant* documents are represented by toolmarks made by the same or another tool, respectively.

4.1 Performance Metrics

For evaluating the performance of our methodology different metrics are used. Firstly, considering the scenario of a forensic expert searching for a linked case. It might be cumbersome to look through hundreds of toolmark images but just searching for n may be feasible. This is captured in a top-n soft criteria which is defined as follows:

$$\text{top-n} = \frac{\sum_{\text{query}} \text{match found in top-n results}}{\text{number of queries}} \tag{3}$$

This means, the score is one if and only if a matching toolmark is found in the first top-n results. Even though this score is very intuitive it has two disadvantages. Firstly, multiple top-n scores have to be combined for an assessment of the performance. Secondly, it does not take into account how many of the relevant toolmarks are found, just if any are found. To fix those shortcomings, the use of the Mean Average Precision (MAP) is proposed. The MAP is calculated as follows [24]:

$$\text{MAP}(Q) = \frac{1}{|Q|} \cdot \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{i=1}^{m_j} precision(R_{ji}) \tag{4}$$

with *information needs* $q_j \in Q$, *relevant documents* $\{d_1, ...d_{m_j}\}$, and R_{jk} the minimal set of ranked retrieval results containing d_k [24]. Further, Precision/Recall plots are used to present the performance of the similarity measure in detail. For comparison with Baiker et al. [4] the F_1 score is used [24].

4.2 Dataset

The NFI dataset published by Baiker et al [4] consists out of 300 toolmarks from 50 different tools. For each tool, toolmarks for $\alpha = 15°$, $30°$, $45°$, $60°$, and $75°$are available. For 10 toolmarks additional 5 toolmarks each at $\alpha = 45°$are provided. However, since a balanced dataset is preferred, these additional $45°$toolmarks are ignore in the NIFT partitionings described below. All toolmarks are available as 2D images, 3D surfaces, or preprocessed 1D profiles extracted from the surfaces. For evaluating the baseline and TripNet the profiles and the 2D images are used, respectively.

Since in contrast to the 1D profiles, the 2D images contain a high variation of translation, scale, and rotation a rough manual correction is performed as a preprocessing step. Further, to increase the number of samples for training and testing TripNet, horizontally flipped versions to the set of 2D images are added. This artificially doubles the number of images to 400. Since invariance to this operation is not desired these images are assigned to a distinct set of an additional 50 tools. The dataset is partitioned into training and testing as follows: All toolmarks of a particular α (including their flipped counterparts for TripNet) are put into the testset; all other toolmarks into the trainingset. The naming of the partitioning is reflected by the toolmarks in the testset, e.g. NFIT 15 contains all toolmarks with $\alpha = 15°$ in the testset.

Furthermore, in order to allow a comparison with [4] the KM 15 vs. KNM and KM 15/30 vs. KNM partitionings presented there are evaluated. These include all comparisons between all matching toolmarks with a difference in α of $15°$, and $15°$or $30°$with all non-matching distances of $\alpha = 45°$, respectively. The additional $\alpha = 45°$ toolmarks are not included in these sets.

To be complete, it should be added that for both the baseline and TripNet the performance for α differences of $0°$was evaluated using the additional $\alpha = 45°$ toolmarks. However, since perfect scores are achieved for this it will not be elaborated in detail.

4.3 Baseline

In Table 2 our baseline is compared to the results published by Baiker et al. [4]. In order to allow a one score comparison, the False Discovery Rate (FDR) and Negative Predictive Value (NPV) given by Baiker et al. were converted into F_1 scores. For the KM 15 vs. KNM evaluation, the difference of 0.002 is neglectable. However, for the dataset containing both $15°$and $30°$comparisons the performance difference increases to about 0.04. This suggests, that our baseline is not as well suited for $\alpha = 30°$ as [4]. Still, the general trend that these methods work well for $\alpha = 15°$ but decrease drastically for $\alpha > 15$ can be observed for both approaches. The performance decline of this approach with increasing α difference is also shown in Figure 5. For this experiment all comparisons were restricted to the given α difference. In order to allow a comparison with α differences of $0°$the additional $\alpha = 45°$ toolmarks were included; all images from the NFI toolmark database were used.

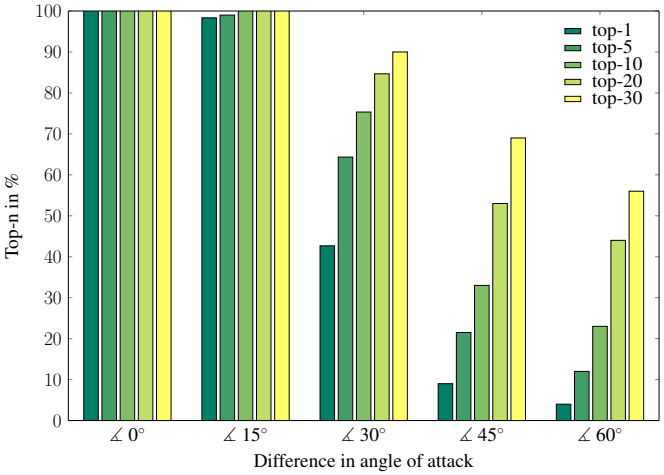

Figure 5: Comparison of the soft-criteria scores achieved by our baseline approach for α differences of $0°$to $60°$.

	Metric	Baiker [4]	Baseline	TripNet
KM 15 vs. KNM	F_1	**0.96**	**0.96**	
KM 15/30 vs. KNM	F_1	**0.79**	0.75	
NFIT 15	MAP		0.47	**0.78**
NFIT 30	MAP		0.69	**0.95**
NFIT 45	MAP		0.70	**0.94**
NFIT 60	MAP		0.56	**0.84**
NFIT 75	MAP		0.35	**0.54**

Table 2: Evaluation results for our baseline and TripNet in comparison with Baiker et al. [4].

This trend is continued for the NIFT datasets. In the NFIT 45 dataset, which similarly contains just comparisons with α differences of $15°$and $30°$, a MAP of 0.70 is achieved. With increasing α difference the MAP drops to 0.47 for NFIT 15 and even 0.35 for NFIT 75 which both contain α differences of $15°$to $60°$. In Figure 6 the steep decline for a recall greater than 0.3 suggests that after correctly identifying the samples with similar α the approach fails to distinguish the remaining toolmarks.

4.4 TripNet

Compared to the baseline it can be seen that the TripNet is better suited to handle α differences greater than $15°$. As shown in Table 2 for NFIT 45 and NFIT 30 a MAP of over 0.9 is achieved which suggests that most of the matching toolmarks are ranked at the top. For NFIT 15 and NFIT 30 the MAP declines slightly to 0.78 and 0.84. However, for NFIT 75 only an MAP of 0.54 is achieved even though the distribution of α differences is the same as for NFIT 15. This could be explained with a degradation of the toolmarks for greater α which is also suggested in [4]. The same can be observed when comparing the result of NFIT 30 with NFIT 60. In Figure 7 these results are shown in detail as Precision/Recall plots.

To investigate the impact of the embedding dimension the Precision/Recall plots are compared in Figure 6. In case of a dimension of 16 the network performs worse than the baseline.

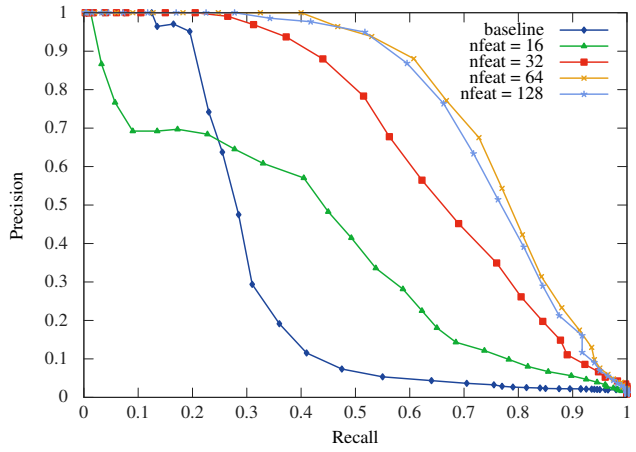

Figure 6: Precision/Recall plot for TripNet comparing different embedding dimensions.

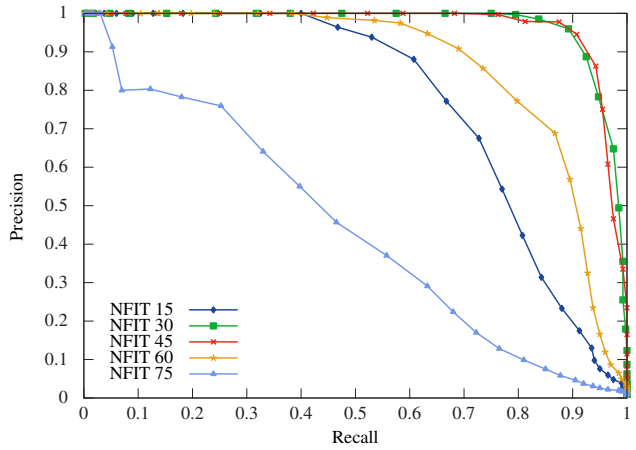

Figure 7: Precision/Recall plot for TripNet comparing different partitionings of the NFI Toolmark dataset.

The sharp drop at a recall of 0.05 suggest that not all toolmarks with an α difference of 15°cannot be distinguished by the network. However, the baseline approach is outperformed by all networks with an embedding dimension of 32 or more. Increasing the embedding dimension to more than 64 does not lead to improved results.

4.5 Computational Effort

The evaluation of the baseline approach was conducted on an Intel i7-5500U CPU using Matlab. On average a distance between two toolmark profiles is calculated in about 3s. Since, NxN (90.000) computations are required it takes 25h to calculate all distances for the whole NFI Toolmark dataset.

In contrast the embedding calculation for TripNet is done in 0.01ms in case the toolmarks are already in memory; otherwise, it takes 1ms. All experiments for TripNet were performed using an NVIDIA Titan X (Maxwell architecture).

4.6 Limitations

It can be seen in Table 2 that the baseline approach is not well suited for distinguishing toolmarks with an α difference of more than 15°. The TripNet handles this situations better, however for extreme cases like NFIT 75 the results are still unsatisfactory. Furthermore, the performance of the network for toolmarks of unseen tools, i.e. tools that are not in the trainingset, was not evaluated.

Additionally, for the current network and dataset the manual translation, rotation and scale correction is essential since the performance degrades significantly to a MAP of just 0.42 for NFIT 15. This drop in performance does not occur when the preprocessed 1D profiles are used although this significantly impairs the network since no random crops can be extracted for training. Therefore, the ability of the network to adapt to variations in the data is severely limited. In this case the MAP drops from 0.78 to 0.67.

5 Conclusion

As shown a main challenge for matching striated toolmarks is to handle differences in angle of attack. In this paper two approaches, an elastic shape matching and a neural network based TripNet, were proposed. Even though a perfect score is achieved by our elastic shape matching baseline when comparing toolmarks made with the same α, it is clearly is not suited for differences of more than 15°; this could however be improved by using a registration scheme similar to [4]. Further, due to the high computational demands of about 3s per comparison this approach is restricted to small toolmark databases in environments without time constraints.

Even though the NFI Toolmark dataset is fairly small the performance achieved by TripNet is promising. As shown the network is able to adapt to α differences of 15 °to 60°achieving an MAP of 0.78 for the NFIT 15 partitioning. For α differences of 15°to 45°in the NFIT 30 partitioning a MAP of 0.95 is achieved. Still, especially for the most challenging NFIT 75 dataset, there is still room for improvement. However, preprocessing is necessary as its robustness to translation, rotation and scaling is limited. For future work this could be improved using bigger trainingsets or by augmenting the input samples to introduce artificial variation. Additional work should also be invested to enable matching of partial toolmarks; for instance, a localized approach with a registration schema.

Acknowledgements

This work has been funded by the Austrian security research programme KIRAS of the Federal Ministry for Transport, Innovation and Technology (bmvit) under Grant 850193. We would like to thank the forensic experts of the Criminal Intelligence Service Austria for their help. The Titan X used for this research was donated by the NVIDIA Corporation. This work was supported by *die Buben*.

References

[1] R. Spotts, L. S. Chumbley, L. Ekstrand, S. Zhang, and J. Kreiser, "Optimization of a Statistical Algorithm for Objective Comparison of Toolmarks," *Journal of Forensic Sciences* **60** no. 2, (2015) 303–314.

[2] B. Bachrach, A. Jain, S. Jung, and R. D. Koons, "A Statistical Validation of the Individuality and Repeatability of Striated Tool Marks: Screwdrivers and Tongue and Groove Pliers," *Journal of Forensic Sciences* **55** no. 2, (2010) 348–357.

[3] M. Page, J. Taylor, and M. Blenkin, "Uniqueness in the forensic identification sciencesFact or fiction?," *Forensic Science International* **206** no. 1, (2011) 12–18.

[4] M. Baiker, I. Keereweer, R. Pieterman, E. Vermeij, J. van der Weerd, and P. Zoon, "Quantitative comparison of striated toolmarks," *Forensic Science International* **242** (2014) 186–199.

[5] M. Baiker, R. Pieterman, and P. Zoon, "Toolmark variability and quality depending on the fundamental parameters: Angle of attack, toolmark depth and substrate material," *Forensic Science International* **251** (2015) 40–49.

[6] M. Baiker, N. D. Petraco, C. Gambino, R. Pieterman, P. Shenkin, and P. Zoon, "Virtual and simulated striated toolmarks for forensic applications," *Forensic Science International* **261** (2016) 43–52.

[7] W. Chu, R. M. Thompson, J. Song, and T. V. Vorburger, "Automatic identification of bullet signatures based on consecutive matching striae (CMS) criteria," *Forensic Science International* **231** no. 1, (2013) 137–141.

[8] L. S. Chumbley, M. D. Morris, M. J. Kreiser, C. Fisher, J. Craft, L. J. Genalo, S. Davis, D. Faden, and J. Kidd, "Validation of tool mark comparisons obtained using a quantitative, comparative, statistical algorithm.," *Journal of Forensic Sciences* **55** no. 4, (2010) 953–61.

[9] N. D. K. Petraco, H. Chan, P. R. D. Forest, P. Diaczuk, C. Gambino, J. Hamby, F. L. Kammerman, W. Brooke, T. A. Kubic, L. Kuo, G. Petillo, E. W. Phelps, A. Pizzola, and D. K. Purcell, "Application of Machine Learning to Toolmarks - Statistically Based Methods for Impression Pattern Comparisons," tech. rep., NCJRS (239048), 2012.

[10] J. Roth, A. Carriveau, X. Liu, and A. K. Jain, "Learning-based Ballistic Breech Face Impression Image Matching," in *Proceedings of IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8. 2015.

[11] J. Song, J. F. Song, and T. V. Vorburger, "Proposed Bullet Signature Comparisons Autocorrelation Functions using," in *Proceedings of National Conference of Standards Laboratories*. 2000.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86** no. 11, (1998) 2278–2324.

[13] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT," *ArXiv* (2014) , arXiv:1405.5769.

[14] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 539–546. 2005.

[15] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4353–4361. Apr, 2015.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[17] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*. 2015.

[18] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors," *ArXiv* (2016) , arXiv:1601.05030.

[19] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape Analysis of Elastic Curves in Euclidean Spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33** no. 7, (2011) 1415–1428.

[20] E. Hoffer and N. Ailon, "Deep metric learning using Triplet network," *ArXiv* (2014) , arXiv:1412.6622.

[21] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv* (2015) , arXiv:1502.03167.

[22] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Muller, "Neural Networks: Tricks of the Trade," *Springer Lecture Notes in Computer Sciences* (1998) 432.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv* (2012) , arXiv:1207.0580.

[24] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.