IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Retrieval of striated toolmarks using convolutional neural networks

ISSN 1751-9632
Received on 31st March 2017
Revised 22nd June 2017
Accepted on 13th July 2017
E-First on 22nd August 2017
doi: 10.1049/iet-cvi.2017.0161
www.ietdl.org

Manuel Keglevic[1] ✉, *Robert Sablatnig*[1]

[1]*Computer Vision Lab, TU Wien, Favoritenstr. 9/183-2, A-1040 Vienna, Austria*

✉ *E-mail: mkeglevic@caa.tuwien.ac.at*

Abstract: The authors propose TripNet as method for calculating similarities between striated toolmark images. The objective for this system is detecting and comparing characteristics of the tools while being invariant to varying parameters like angle of attack, substrate material, and lighting conditions. Instead of designing a handcrafted feature extractor customised for this task, the authors propose the use of a convolutional neural network. With the proposed system, one-dimensional profiles extracted from images of striated toolmarks are mapped into an embedding. The system is trained by minimising 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 Netherlands Forensic Institute. The system proposed is able to adapt to a large range of angles of attack, achieving a mean average precision of 0.95 for toolmark comparisons with differences in angle of attack of 15°–45°. Furthermore, four different triplet selection approaches are proposed and their effect on the retrieval of toolmarks from a database of unseen tools is evaluated in detail.

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, 4–9] for automatically, and objectively [4], comparing striated toolmarks.

The input for these algorithms are one-dimensional (1D) profiles extracted from either 2D images or 3D surface scans of the striated toolmarks. In Fig. 1, images with superimposed profiles from the same tool at different angles of attack are depicted. After pre-processing, similarity scores are commonly computed using the cross-correlation either globally on the whole profile [4–6] or locally [8]. This approach is also proposed by the National Institute of Standards and Technology (NIST) for comparing ballistic toolmarks [10, 11]. Another similarity measure based on locally normalised squared distances, the so-called *relative distance*, is proposed by Bachrach *et al.* [2].

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 and linear discriminant analysis 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 detecting and comparing individual, class, and sub-class characteristics of the tools [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 an order of magnitude higher than for differences of 15°, i.e. the false discovery rate (FDR) 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

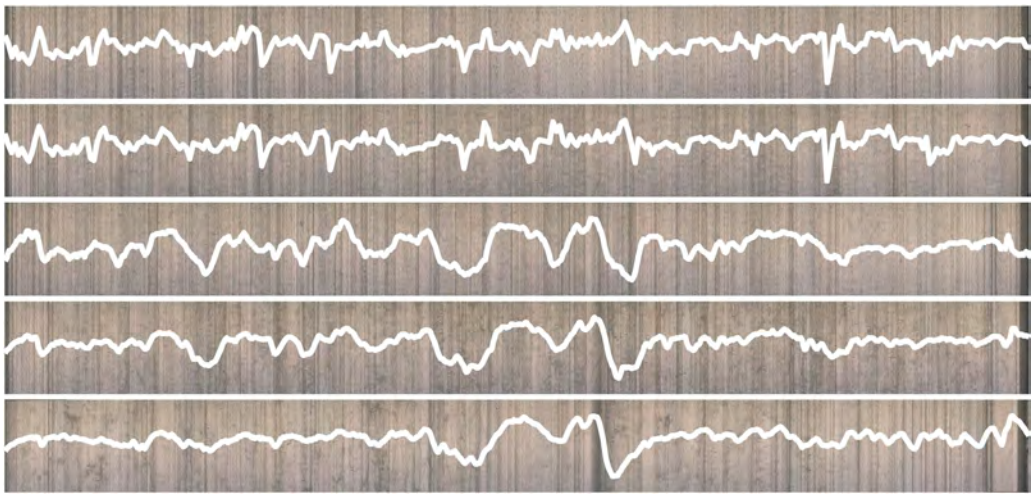

Fig. 1 *Superimposed 1D profiles extracted from 3D surface scans onto 2D images of NFI Toolmarks. All marks were made by the same tool with varying angle of attack; from top to bottom: 15°, 30°, 45°, 60°, and 70°*

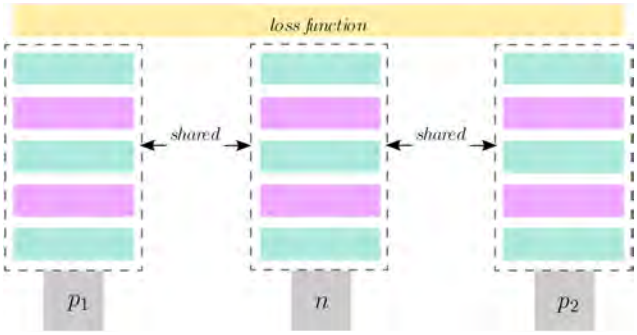

Fig. 2 *Triplet architecture*

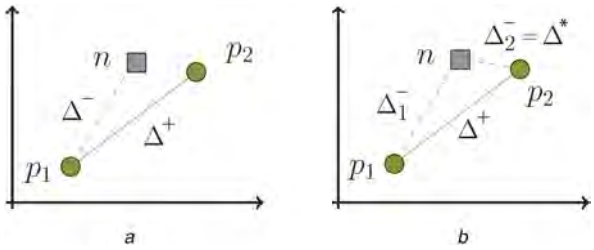

Fig. 3 *SoftMax ratio* ***(a)*** *Compared with SoftPN,* ***(b)*** *[17]*

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. Further, the complicated problem of distinguishing between class, sub-class, and individual characteristics is circumvented this way. For comparing images, Chopra *et al.* [13] 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 L_1 norm. A similar architecture is applied by Zagoruzkoa and Komadaski [14] for matching local image patches. Schroff *et al.* [15] propose the use of a triplet loss function minimising the distance between an anchor and a positive (match) while maximising the distance between the anchor and a negative sample (non-match). In like manner, similar architectures are used to distinguish faces [16] and local image patches [17].

In this paper, we propose the use of a CNN called 'TripNet' for matching toolmark profiles. In contrast to other approaches like [15], we use profiles extracted from 2D images and do not rely on 3D surface scans. In this way, 'TripNet' can be applied to images captured under a forensic comparison microscope. To allow a fast computation of similarities, the triplet loss function described by Balntas *et al.* [17] is applied. Hence, 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. Four different approaches for selecting triplets are presented. Our evaluation is based on the NFI database of 300 striated screwdriver toolmarks published by Baiker *et al.* [4]. In addition to the retrieval of matching tools from an annotated and trained database, the retrieval of unseen tools is also evaluated. Furthermore, the effects of the different triplet selection approaches are shown in detail.

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 it 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 is based on the elastic shape metric proposed by Srivastava *et al.* [18]. This approach is publicly available (http://ssamg.stat.fsu.edu/software) and requires no parameter evaluation. 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 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 minimising geodesic between their point representations in *pre-shape-space*. Since this *pre-shape-space* is not invariant to rotation and re-parameterisation, an additional optimisation step is performed afterwards to compute the distances in *shape-space*. The methodology is described in detail in [18].

This approach is directly applied to the NFI Toolmark profiles after downsampling to 800 points, which corresponds to the minimal wavelength used by Baiker *et al.* [4]. The extensive pre-processing pipeline applied to the NFI profiles is described in [4] and includes cropping, stitching, alignment, global shape removal, and noise reduction.

3 TripNet

Our proposed neural network TripNet is based on the work of Balntas *et al.* [17]. The architecture is depicted in Fig. 2. Similar to siamese networks [13], there are multiple branches 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. The loss function minimises the Euclidean distance between matching samples, and therefore the L_2 norm can be used to measure distances in the embedding. Consequently, efficient algorithms for calculating L_2 distances can be applied [17]. Additionally, the storage requirements are directly controlled by changing the dimension of the embedding.

3.1 Triplet loss

In contrast to other triplet loss functions like the SoftMax Ratio proposed by Hoffer and Ailon [19], which only takes one negative distance into account, all three distances between the samples are used in [17]

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

with the triplet $T = \{x_{p_1}, x_{p_2}, x_n\}$ and the embedding $f(x)$. Instead of forcing the distance Δ^+ just to be smaller than Δ_1^-, it is forced to be smaller than $\Delta^* = \min(\Delta_1^-, \Delta_2^-)$. The difference is illustrated in Fig. 3.

The loss is then defined as [17]

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

which is implemented using a Softmax layer and the mean square criterion. The selection of training samples is simplified by this approach, as soft negative mining is performed implicitly [17].

3.2 CNN architecture

The architecture of the CNN is depicted in Table 1. As our input samples are profiles (or profile segments), the convolutional and pooling layers have 1D input regions.

Each convolutional layer is followed by batch normalisation to decrease the dependency on input normalisation and initialisation of the network [20]. 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 1×5 convolutions and 1×3 pooling with 64 feature maps in the first convolution and 32 in the second. In contrast to [17], rectified linear units (ReLU) [21] and average pooling [12] are used, since this setup performs more desirable. 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 [22] layer is added at the end with a probability of 0.5.

3.3 Triplet selection

Since the samples presented to the CNN during training define which features are learned, the selection of these samples is crucial. Firstly, the trainingset must contain enough samples from different classes to allow an identification of the distinguishing characteristics by the CNN. Secondly, class and sub-class variations must be represented to improve the robustness. Especially when dealing with small datasets, improvements can be made by either artificially increasing the variations in those categories, or by carefully selecting the training samples (similarly to negative mining [17]).

In the case of TripNet, the loss function defined in (2) minimises the loss by separating the positive samples from the negatives in the embedding. Since the weights of all three branches of the CNN are shared, the feature extractors learned for all three samples in a triplet are the same; i.e. equal samples lead to equal representations in the embedding. This leads to the following reasoning: firstly, in case a local characteristic is represented in all three samples, this characteristic cannot improve the loss and is therefore suppressed by the CNN. This allows the introduction of artificial variations to increase the robustness because the network learns that these variations are not distinctive. Secondly, if a local characteristic is just present in the positive samples and not in the negative, the CNN uses this local characteristic to separate the samples. However, this does not only apply to the differences in the individual characteristics we want to detect, but also to unwanted characteristics like varying lighting conditions, small differences in camera angle, artificial data augmentations, and so on. For instance: considering data augmentation applied just to the positive samples. The trivial solution for separating the samples, and therefore minimising the loss, is using features, which detect the local characteristics of the augmentation artefacts itself. Even though this in turn would lead to a rapid decline of the loss function during training, the results would not improve at all since the variation itself does not represent natural occurring variations we want to capture.

As a result of these considerations, the following triplet selection and data augmentation strategies are implemented to implicitly define which characteristics are relevant and which should be suppressed by the CNN:

Full profiles: During training, random vertical crops are taken to increase the variability of the samples. For evaluation, only centre crops are used to ensure reproducibility.

Permutated profiles: The training samples are chosen similarly to the full profiles. However, the negative sample and the positive samples are permuted randomly with the same factor. Since the position of local characteristics is a defining factor for a tool, two new artificial tools are created during training with both matching and non-matching toolmark profiles in this way. It is crucial that the permutation is performed simultaneously on the whole triplet. In case the negative sample alone is permuted independently, the position of the stitching artefact, i.e. the seam, will be learned by the CNN in order to distinguish positive and negative samples. In case all three samples are permuted independently, the exact position of a local characteristic on the profile is suppressed and can no longer be used to distinguish toolmarks. The idea is to increase the number of possible triplets by moving the individual characteristics to different positions and thus instructing the CNN not only to detect local characteristics but also to observe their position on the profiles. Similarly to the full profiles, centre crops without permutations are used for evaluation.

Segments: Instead of training the whole profile, profile segments are randomly cropped from the input images. Since this introduces no seams like they are by the permutated profiles described above, it can be done for the positive samples and the negative samples independently. The architecture of the CNN branches remains the same. Due to the smaller input however, the overall number of parameters of the CNN is decreased. The length of the segments is evaluated in Section 4.4. To compute the similarity value of two profiles during evaluation, a sliding window approach is applied to profiles cropped from the centre of the image.

Patches: The pre-processing pipeline proposed by Baiker *et al.* [4] for the NFI Toolmark profiles includes an averaging along the x-axis. Therefore, to investigate the influence of adding a second dimension for noise reduction, random square patches are cropped from the input images. Similarly to the profile segments, the positive samples are cropped simultaneously, i.e. at the same position. However, preliminary tests show that the negative sample must be extracted from one of the positive images to prohibit trivial solutions like different lighting conditions, contrast, and so on. To ensure that the negative sample does not overlap with the positives, a safety distance is implemented. Furthermore, to counteract learning small angular variations in the camera angle horizontal flips (reflected along the central vertical axis) of the samples are performed randomly. Since the input patches are 2D, the architecture of the CNN branches depicted in Table 1 is changed accordingly, i.e. 5×5 and 3×3 regions are used for the convolutional and max pooling layers, respectively. Similarly to the profile segments, sliding windows in the centre of the toolmark images are used to compute similarity scores during evaluation.

3.4 Implementation details

The training and evaluation of TripNet is implemented in Torch (https://github.com/torch/torch7). Similarly to the processed 1D profiles used for the baseline, the 2D images are downscaled to a height of 800 pixels, which leads to a resolution of about 100 pixels per millimetre. The triplet creation is done on-line, i.e. not created beforehand but during training. Min/max-normalisation and mean pixel subtraction is performed as a pre-processing step.

Similarity scores are computed by calculating the L_2 distance between the representations of the toolmarks in the embedding. For the profile segments and patches, a sliding window is used to compute multiple representations, i.e one for each segment or patch, from top to bottom. The sum of the pairwise distances of the corresponding representations is then used as the distance measure between two toolmarks. The stepsize is set to 1/16 of the height of the segment or patch.

The optimisation is done using stochastic gradient decent with a learning rate of 0.01, weight decay of 10^{-4}, and momentum of 0.9.

4 Evaluation

As opposed to other works on toolmarks [2, 4], 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 *non-relevant* documents from a *document collection* [23]. 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 *non-relevant* documents are represented by toolmarks made by the same or another tool, respectively.

4.1 Performance metrics

For evaluating the performance of our methodology, various metrics are used. Firstly, considering the scenario of a forensic expert searching for a linked case, it would be cumbersome to look

Table 1 Architecture of the CNN branches

Layer #	Description
1	spatial convolution(1,5) → 64
2	spatial batch normalisation
3	ReLU
4	average pooling(1,3)
5	spatial convolution(1,5) → 32
6	spatial batch normalisation
7	ReLU
8	average pooling(1,3)
9	dropout
10	linear → nfeat
11	Tanh

Table 2 Results for our baseline and TripNet in comparison with [4]

	Metric	Baiker [4]	Baseline	TripNet full profile
KM 15 versus KNM	F_1	**0.96**	**0.96**	
KM 15/30 versus 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**

Bold values: Maximum for each row/dataset

through hundreds of toolmark images but only searching through n may be feasible. This is captured in a top-n soft criterion, which is defined as follows:

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

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 whether any are found at all. To correct for those shortcomings, the use of the mean average precision (MAP) is proposed. The MAP is calculated as follows [23]:

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

with information needs $q_j \in Q$, the set of all information needs Q, relevant documents $\{d_1, \ldots, d_{m_j}\}$, and R_{ji} the minimal set of ranked retrieval results containing d_i [23]. In the case of toolmark images, q_j can be formulated as 'find images with toolmarks made by the same tool as the supplied image' and d_i as 'image with toolmarks made by the same tool'. The retrieval results for each q_i are ranked by similarity score. Each R_{ji} then contains d_i and all other images, which are more similar to the supplied image than d_i. A perfect score of 1.0 is achieved when all d_i are ranked at the top, and thus all R_{ji} contain only relevant documents.

To present the performance of the similarity measure in detail, precision/recall plots are used additionally to the MAP. Furthermore, for comparison with Baiker *et al.* [4] the F_1 score is used [23].

4.2 Dataset

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

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

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 (known match) 15 versus KNM (known non-match) and KM 15/30 versus 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.

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

4.3 Baseline

In Table 2, our baseline is compared with the results published by Baiker *et al.* [4]. In order to allow a one-score comparison, the FDR and negative predictive value given by Baiker *et al.* were converted into F_1 scores. For the KM 15 versus KNM evaluation, the difference of 2 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.

This trend is continued for the NFIT 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 $15°$–$60°$. In Fig. 4, the steep decline for a recall >0.3 suggests that after correctly identifying the samples with similar α, the approach fails to distinguish the remaining toolmarks.

Since this approach computes similarities without prior training, similar performance is achieved on the SPLIT datasets as shown in Table 4. The slight improvements in MAP are due to the fact that only tools with odd numbers are in the testset, and therefore the total number of images is reduced.

4.4 TripNet

The NFIT datasets indicate that the basic TripNet with full profiles is better suited to handle α differences $>15°$ than the baseline. 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 60, the MAP declines

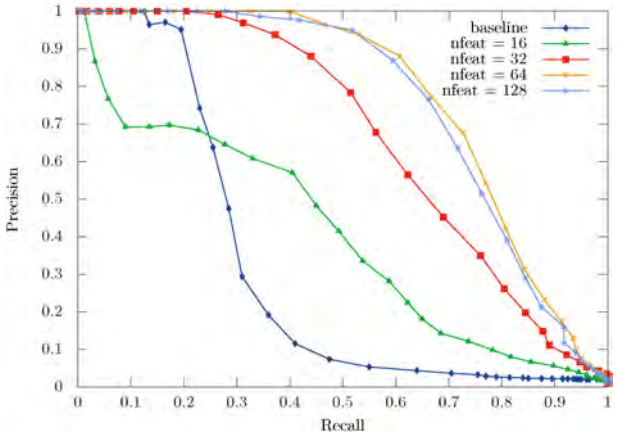

Fig. 4 *Precision/recall plot for TripNet with full profiles comparing different embedding dimensions*

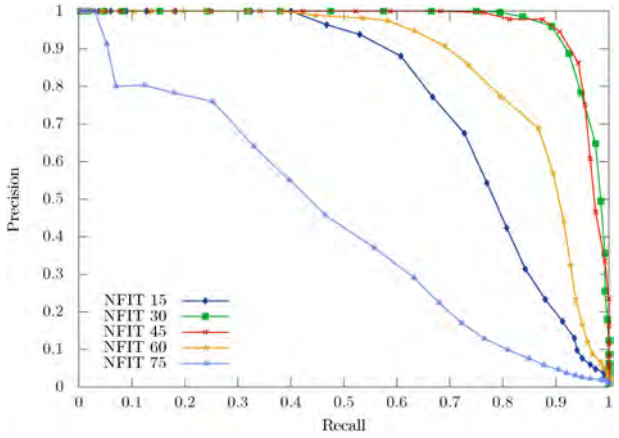

Fig. 5 *Precision/recall plot for TripNet with full profiles comparing different partitionings of the NFI Toolmark dataset*

Table 3 Results on the NFIT datasets filtered by α difference

α difference	−15	+15	−30	+30	−45	+45	−60	+60
NFIT 15		0.88		0.82		0.81		0.79
NFIT 30	**0.99**	**0.97**		0.95		**0.94**		
NFIT 45	0.98	0.96	**0.98**	**0.96**				
NFIT 60	0.92	0.88	0.86		**0.85**			
NFIT 75	0.69		0.58		0.57		0.61	

Bold values: Maximum for each column/alpha difference

slightly to 0.78 and 0.84. However, for NFIT 75 a MAP of only 0.54 is achieved even though the distribution of α differences is the same as for NFIT 15. This can be explained by a degradation of the toolmarks for greater α, which is also suggested in [4] and indicated in Table 3. The same can be observed when comparing the result of NFIT 30 with NFIT 60. In Table 3, the results are filtered by α differences to separate the influence of angle of attack in the testset from the α difference in the retrieval results. This demonstrates that overall the retrieval of toolmarks is more challenging for NFIT 15, NFIT 60, and NFIT 75 compared with NFIT 30 and NFIT 45 even for an α difference of only 15°. However, this table also shows that even for NFIT 15, a MAP of 0.79 can be achieved when only the toolmark images with an α of 75° are considered for retrieval.

In Fig. 5, the retrieval results are shown in detail as precision/recall plots. To investigate the impact of the embedding dimension, the precision/recall plots are compared in Fig. 4. In the case of a dimension of 16, the network performs worse than the baseline. The sharp drop at a recall of 0.05 suggests that not all toolmarks with an α difference of 15° can 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.

Concerning the SPLIT datasets, the TripNet with full profiles performs significantly worse than the baseline. As shown in Table 4 for SPLIT 15, only a MAP of 0.27 is achieved compared with a MAP of 0.55 for the baseline. This means that even though the approach is able to successfully separate toolmark images from different tools, it is not able to generalise to unseen tools. Fig. 6 shows a precision/recall plot in which the different triplet selection methods proposed in Section 3.3 are compared using the SPLIT 15 dataset. Since the dataset is fairly small, the first method introduces random permutations to the profiles in order to extrapolate the training data. Even though this improves the MAP from 0.27 to 0.36, it still performs worse than the baseline. As expected, the increased MAP of 9% shows that decoupling the local characteristics of the toolmarks from their position during training is advantageous since this artificially increases the trainingset and the CNN learns that not only the presence, but also the position of a local characteristic is important. Using randomly extracted 1×48

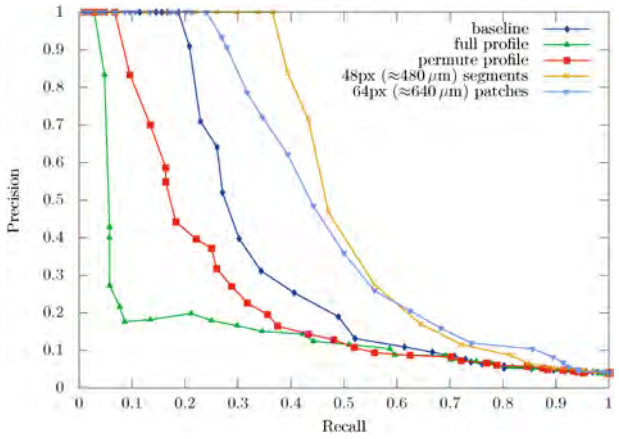

Fig. 6 *Precision/recall plot comparing the proposed triplet selection approaches using the SPLIT 15 dataset*

pixel ($\simeq 480\,\mu$m) segments, and thus decoupling the local characteristics from the position on the profile completely during training, leads to further improvements with a MAP of 0.56. Even though the MAP is only slightly better than the baseline, the recall/precision plot shows that the resulting similarity measure is more distinctive. The results filtered by α difference in Table 5 indicate that for unseen tools, this method only works well for differences of 15° with MAPs between 0.81 and 1.00. However, for α differences of 30° MAPs of at least 0.66 are achieved in case toolmarks with an α of 75° are not considered; comparisons including toolmarks with an α of 75° perform significantly worse than all others. Using patches instead of segments in order to improve robustness leads to similar performance but does not offer measurable improvement in MAP and the precision/recall plot is slightly worse than with segments.

In Fig. 7, the impact of the segment size on the performance is depicted using the SPLIT 15 dataset. Overall, the performance is similar with a MAP ranging from 0.52 to 0.56. The best performance is achieved with 1×48 pixel segments. Table 4 shows that this approach, using segments with 1×48 pixel, achieves at least the same performance as the baseline and can lead to a performance increase of up to 11% MAP depending on the dataset. Even though the results on the SPLIT datasets are not as promising as on the NFIT datasets, a MAP of over 0.70 can be achieved for SPLIT 30, SPLIT 45, and SPLIT 60 with α differences of 15°–45°. In Fig. 8, the precision/recall plots for the different datasets are compared in detail.

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 3 s. Since NxN (9000)

Table 4 Results for the retrieval of toolmark images of unseen tools

	Metric	Baseline	TripNet full profile	TripNet segments
SPLIT 15	MAP	0.55	0.27	**0.56**
SPLIT 30	MAP	**0.75**	0.35	**0.75**
SPLIT 45	MAP	0.75	0.31	**0.77**
SPLIT 60	MAP	0.61	0.23	**0.72**
SPLIT 75	MAP	0.41	0.16	**0.44**

Bold Values: Maximum for each row/dataset

Table 5 Results on the SPLIT datasets filtered by α difference

α difference	−15	+15	−30	+30	−45	+45	−60	+60
Split 15		**1.00**		0.66		**0.45**		0.35
Split 30	0.98	**1.00**		**0.71**		0.38		
Split 45	**1.00**	0.98	0.68	0.55				
Split 60	0.98	0.81	**0.69**		**0.48**			
Split 75	0.82		0.57		0.32		0.35	

Bold values: Maximum for each column/alpha difference

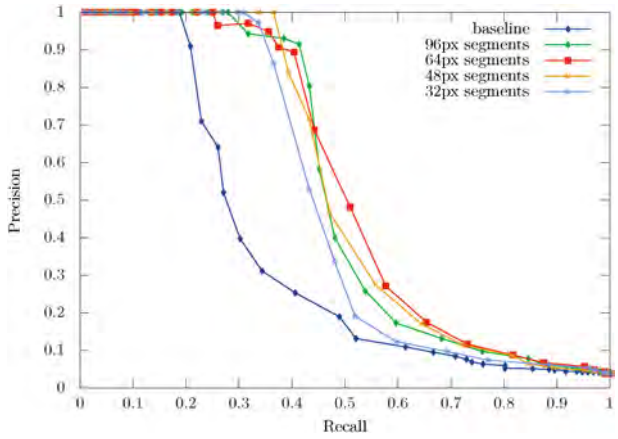

Fig. 7 *Precision/recall plot comparing the effect of varying segment length using the SPLIT 15 dataset*

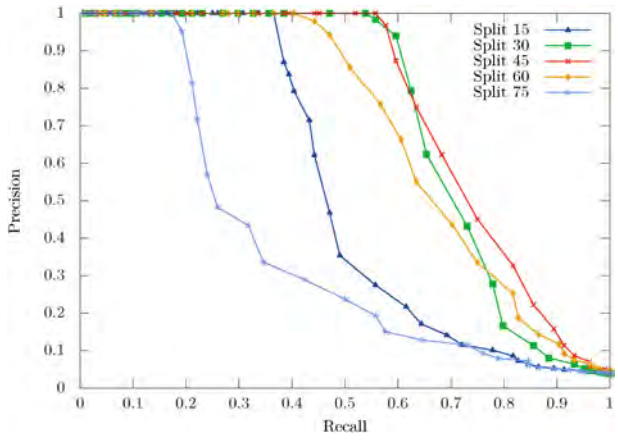

Fig. 8 *Precision/recall plot comparing the performance of the TripNet with 1 × 48 segments on the SPLIT datasets*

computations are required, it takes 25 h to calculate all distances for the whole NFI Toolmark dataset.

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

4.6 Limitations

It can be seen in Table 2 by the performance drop from an F_1 score of 0.96–0.75 from KM 15 versus KNM to KM 15/30 versus KNM, that the baseline approach is not well suited for distinguishing toolmarks with an α difference of more than 15°. The TripNet handles these situations better; however, for extreme cases like NFIT 75, the results are still unsatisfactory. Furthermore, when comparing Table 2 with Table 4, the performance of TripNet for toolmarks of unseen tools still leaves room for improvement. Since the dataset is fairly small, dividing the profiles into segments is necessary to improve the performance. This however introduces a handcrafted sliding window approach for combining the segment representations in the embedding, which is not advanced enough to capture additional higher level information encoded in the profiles.

Additionally, 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 pre-processed 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 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 3 s 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 demonstrated, the network is able to adapt to α differences of 15°–60° achieving a MAP of 0.78 for the NFIT 15 partitioning. For α differences of 15°–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. Even though the performance of the TripNet with full profiles for toolmarks of unseen tools cannot compete with the above results, by using profile segments instead a MAP of 0.75 can be achieved on the SPLIT 30 dataset with α differences of 15°–45°. Furthermore, the baseline approach can be outperformed in this task by a MAP of up to 9%, although the calculations can be performed significantly faster; i.e. the computation of all distances in the SPLIT 15 dataset takes about 20 s instead of several hours. For future work, the performance could be improved by replacing the sliding window approach used for the distance calculations. This could for instance be done by pre-training the lower layers of the full profile TripNet using profile segments. Furthermore, the creation of an increased dataset

would be beneficial to the performance and provide means for a more detailed evaluation. Additional work should also be invested to evaluate how well TripNet with segments is suited for matching of partial toolmarks.

6 Acknowledgments

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. The authors 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.

7 References

[1] Spotts, R., Chumbley, L.S., Ekstrand, L., *et al.*: 'Optimization of a statistical algorithm for objective comparison of toolmarks', *J. Forensic Sci.*, 2015, **60**, (2), pp. 303–314

[2] Bachrach, B., Jain, A., Jung, S., *et al.*: 'A statistical validation of the individuality and repeatability of striated tool marks: screwdrivers and tongue and groove pliers', *J. Forensic Sci.*, 2010, **55**, (2), pp. 348–357

[3] Page, M., Taylor, J., Blenkin, M.: 'Uniqueness in the forensic identification sciences-Fact or fiction?', *Forensic Sci. Int.*, 2011, **206**, (1), pp. 12–18

[4] Baiker, M., Keereweer, I., Pieterman, R., *et al.*: 'Quantitative comparison of striated toolmarks', *Forensic Sci. Int.*, 2014, **242**, pp. 186–199

[5] Baiker, M., Pieterman, R., Zoon, P.: 'Toolmark variability and quality depending on the fundamental parameters: angle of attack, toolmark depth and substrate material', *Forensic Sci. Int.*, 2015, **251**, pp. 40–49

[6] Baiker, M., Petraco, N.D.K., Gambino, C., *et al.*: 'Virtual and simulated striated toolmarks for forensic applications', *Forensic Sci. Int.*, 2016, **261**, pp. 43–52

[7] Chu, W., Thompson, R.M., Song, J., *et al.*: 'Automatic identification of bullet signatures based on consecutive matching striae (CMS) criteria', *Forensic Sci. Int.*, 2013, **231**, (1), pp. 137–141

[8] Chumbley, L.S., Morris, M.D., Kreiser, M.J., *et al.*: 'Validation of tool mark comparisons obtained using a quantitative, comparative, statistical algorithm', *J. Forensic Sci.*, 2010, **55**, (4), pp. 953–961

[9] Petraco, N.D.K., Chan, H., Forest, P.R.D., *et al.*: 'Application of Machine Learning to Toolmarks – Statistically Based Methods for Impression Pattern Comparisons' NCJRS (239048), 2012

[10] Roth, J., Carriveau, A., Liu, X., *et al.*: 'Learning-based ballistic breech face impression image matching'. Proc. IEEE Seventh Int. Conf. on Biometrics Theory, Applications and Systems (BTAS), 2015, pp. 1–8

[11] Song, J., Song, J.F., Vorburger, T.V.: 'Proposed bullet signature comparisons autocorrelation functions using'. Proc. National Conf. of Standards Laboratories, 2000

[12] Lecun, Y., Bottou, L., Bengio, Y., *et al.*: 'Gradient-based learning applied to document recognition', *Proc. IEEE*, 1998, **86**, (11), pp. 2278–2324

[13] Chopra, S., Hadsell, R., LeCun, Y.: 'Learning a similarity metric discriminatively, with application to face verification'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), vol. **1**, 2005. pp. 539–546

[14] Zagoruyko, S., Komodakis, N.: 'Learning to compare image patches via convolutional neural networks'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4353–4361

[15] Schroff, F., Kalenichenko, D., Philbin, J.: 'FaceNet: a unified embedding for face recognition and clustering'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015

[16] Parkhi, O.M., Vedaldi, A., Zisserman, A.: 'Deep face recognition'. Proc. British Machine Vision Conf. (BMVC), 2015

[17] Balntas, V., Johns, E., Tang, L., *et al.*: 'PN-Net: conjoined triple deep network for learning local image descriptors', ArXiv, 2016

[18] Srivastava, A., Klassen, E., Joshi, S.H., *et al.*: 'Shape analysis of elastic curves in Euclidean spaces', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (7), pp. 1415–1428

[19] Hoffer, E., Ailon, N.: 'Deep metric learning using triplet network', ArXiv, 2014

[20] Ioffe, S., Szegedy, C.: 'Batch normalization: accelerating deep network training by reducing internal covariate shift', ArXiv, 2015

[21] LeCun, Y., Bottou, L., Orr, G.B., *et al.*: '*Neural networks: tricks of the trade*' (Springer Lecture Notes in Computer Sciences, 1998), p. 432

[22] Hinton, G.E., Srivastava, N., Krizhevsky, A., *et al.*: 'Improving neural networks by preventing coadaptation of feature detectors', ArXiv, 2012

[23] Manning, C.D., Raghavan, P., Schütze, H.: '*Introduction to information retrieval*' (Cambridge University Press, 2008)