Unsupervised Learning of Fingerprint Rotations

Patrick Schuch[1], Jan Marek May[2], Christoph Busch[3]

Abstract: The alignment of fingerprint samples is a preprocessing step in fingerprint recognition. It allows an improved biometric feature extraction and a more accurate biometric comparison. We propose to use Convolutional Neural Networks for estimation of the rotational part. The main contribution is an unsupervised training strategy similar to Siamese Networks for estimation of rotations. The approach does not need any labelled data for training. It is trained to estimate orientation differences for pairs of samples. Our approach achieves an alignment accuracy with a mean absolute deviation 2.1° on data similar to the training data, which supports the alignment task. For other datasets accuracies down to 6.2° mean absolute deviation are achieved.

Keywords: fingerprint recognition, machine learning, alignment, unsupervised learning.

1 Introduction

Despite the fact that fingerprint recognition is a mature and widely deployed technology, it still can be improved or extended to uprising demands. Some aspects in fingerprint recognition can benefit from *aligning* fingerprint samples to a common orientation and positioning. Both concerns can be tackled separately [Ma09]. This works focusses on the orientation part of the alignment. There are two categories of alignments: alignments common for all fingerprints and those shared for all samples of the same finger. The former is the more general approach. The latter allows a higher degree of freedom in the alignment, since each fingerprint may have its individual alignment. Such an alignment can be beneficial in many cases. Most important, biometric comparison algorithms may benefit and especially speed up since they do not have to deal with large rotations between compared samples.

A trivial orientation alignment is the upright position in the direction of the fingertip. But finding this trivial orientation is actually far from being trivial. Usually, fingerprint alignment works on *focal points* [RA00]. Detection of such points is challenging, if the fingerprint quality is low or there is only a partial print without any focal points in it. Another serious challenge in fingerprint alignment is the fact there is no *ground truth* for the alignment, i.e. you never know for sure what the right alignment is. No assumptions on *initial rotations* of fingerprint samples can be made. Any manual labelling is prone to inaccuracies and lacks reproducibility.

Deep learning has provided quite impressive solutions in many domains of digital image processing and pattern recognition in the last years. We propose to use *Convolutional Neural Networks* (CNN) for the task of fingerprint alignment. Our main contribution is

[1] NTNU, NBL Norwegian Biometrics Lab, Gjøvik, Norway, patrick.schuch2@ntnu.no
[2] Dermalog Identification Systems GmbH, Hamburg, Germany, marek.may@dermalog.com
[3] NTNU, NBL Norwegian Biometrics Lab, Gjøvik, Norway, christoph.busch@ntnu.no

a training strategy. Similar as in *Siamese Networks* the CNN learns a rotational distance between two arbitrarily rotated instances of a single fingerprint sample. As the approach is not dependant on any ground truth data, the training can be done unsupervised. Large amounts of unlabelled data in turn allow to train the CNN. The training finally yields an orientation assignment, which is individual to each finger.

The rest of this paper is organized as follows: Section 2 provides an overview on related work. Our approach will be explained in Section 3. Section 4 explains the experiments, which were carried out to test the proposed approach. Section 5 concludes the findings. An outlook on future work can be found in Section 6.

2 Related work

Several approaches for the task of fingerprint pose estimation have been proposed in the past. Sood and Kaur provide an overview on alignment methods in fuzzy vault [SK14].

Markert et al. proposed to use the almost parallel ridges above creases for alignment, as those ridges usually all have similar directions in every fingerprint [Ma18].

Most of the approaches make use of the orientation field of a fingerprint. The orientation field is a representation for the local orientations of the ridges in a fingerprint. If no original fingerprint sample is available, the orientation field can also be estimated for extracted fingerprint minutiae [Kr14].

Some approaches work directly on the orientation fields. Yang et al. proposed to learn dictionaries of orientation field patches [YFZ14]. They used the orientations fields to perform a pose estimation for the fingerprint. Hotz proposed to extract an intrinsic coordinate system based on a longitudinal axis [Ho09]. The longitudinal axis can be found by searching for symmetries in the orientation field. This axis could also be used for a rotational alignment.

Other approaches extract distinctive points from the orientation field. So-called *singularities* (cores and deltas) can be used for an alignment [Ja00]. For fingerprints lacking singularities *focal point* can be defined as those point with the highest curvature [RA00]. Nagar et al. and Zhang et al. also used points of maximum curvature as reference points for an alignment [NRV10] [ZFH14]. Li et al. proposed to use isosceles triangles for alignment [LBY14]. Isosceles triangles are placed on the ridges. The approach makes use of local symmetries near focal points. Liu et al. proposed a multi-scale approach for detection of the focal points from orientation fields [LJK05]. Tams proposed to extract reference points from the fingerprint orientation field [Ta13]. Jain and Minut used kernel curves to describe the flow of the fingerprint ridges [JM02]. Those kernel curves describe the behaviour of the ridges around the focal points. Best fitting kernel curves can be used for an alignment to typical patterns. Li et al. proposed to estimate an alignment by topological structures around cores [Li08]. Detection of singularities and focal points depends on an accurate extraction of the orientation field. Such a detection may fail for partial fingerprint samples.

Merkle et al. proposed to use the shape of the fingerprint sample [Me10]. The outline of the fingerprint sample usually can be approximated by an ellipse. This ellipse is aligned to its principal axis. Yang et al. proposed to align small neighbourhoods of minutiae [YB09].

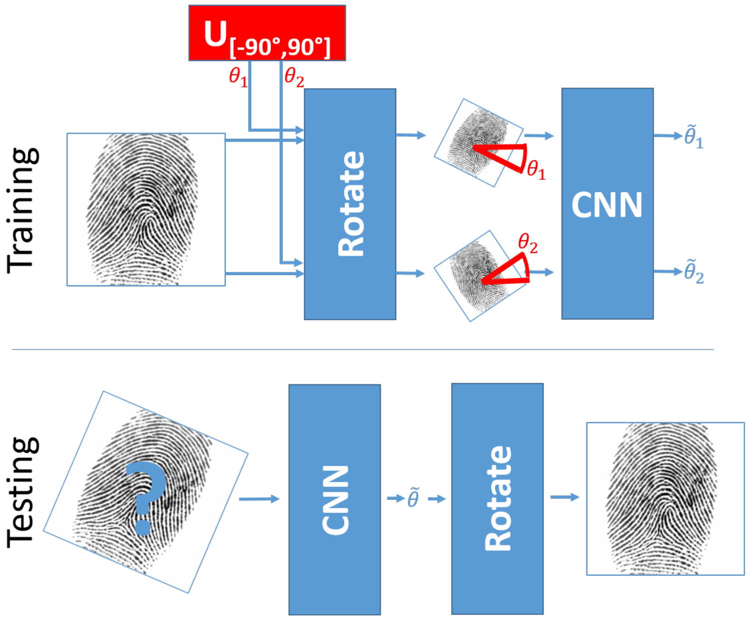


Fig. 1: During training a fingerprint sample is rotated by two random angles θ_1 and θ_2. The CNN estimates rotations $\tilde{\theta}_1$ and $\tilde{\theta}_2$, so that differences (θ_1-θ_2) and ($\tilde{\theta}_1$-$\tilde{\theta}_2$) are as similar as possible. During testing estimation $\tilde{\theta}$ can be used for rotational alignment.

There is also an approach, which uses techniques from the domain of Deep Learning. Ouyang et al. proposed to use a variation of a *Region-based Convolutional Network* (R-CNN), which have originally been proposed to the task of object detection [Ou17]. The authors compared the estimated poses to manually labelled ground truth poses. They have also found positive effects of alignment when applying *Fingerprint Indexing*.
There are further approaches in Deep Learning for the task of general alignment, e.g. *Spatial Transformer Networks* proposed by Jaderberg et al. [Ja15]. Those can learn transformations of sampling grids for given tasks. The original image is then sampled at the transformed sampling grid. Any transformation, which helps solving the given task, can be learned in this approach. Our approach does not need a given task for training.

3 Our Approach

CNNs for Estimations of Rotations We will exploit a training strategy called *Siamese Network* [Br94] for the task of learning fingerprint alignment. Each training input consists of a pair of randomly rotated instances of a single fingerprint sample (see Figure 1). The CNN estimates a rotation for each instance. The CNN shall learn to assign estimations in such a way, that the difference between the two estimated values is close to the difference of the two actual rotations. The main advantage of this approach is, that no expert based ground truth rotational information is necessary at all. This allows to use any unlabelled

#	Layer	Output
0	Input (192×192×1)	(192×192×1)
1	ConvLayer (32×5×5×1)	(192×192×32)
2	BatchNorm	(192×192×32)
3	PReLU	(192×192×32)
4	MaxPooling (4×4)	(48×48×32)
5	ConvLayer (32×5×5×32)	(48×48×32)
6	BatchNorm	(48×48×32)
7	PReLU	(48×48×32)
8	MaxPooling (4×4)	(12×12×32)
9	ConvLayer (32×5×5×32)	(12×12×32)
10	BatchNorm	(12×12×32)
11	PReLU	(12×12×32)
12	MaxPooling (4×4)	(3×3×32)
13	Flatten	(288)
14	Dense (256×288)	(288)
15	PReLU	(256)
16	Dense (256×256)	(256)
17	PReLU	(256)
18	Dense (181×256)	(181)
19	Softmax	(181)

Tab. 1: CNN's architecture. The final output can be interpreted as probabilities for different alignment angles.

fingerprint sample for training. For testing the trained CNN estimates the rotation for a given input fingerprint sample (see Figure 1). This estimation can be used to align the input fingerprint sample by a corresponding rotation.

Architecture There has been no extensive optimization of any hyperparameters. The entire model assembles seven different types of layers: *Convolutional* layers (ConvLayer), *Parametric Rectified Linear Units* (PReLu), *Batch Normalization* layers (BatchNorm), *Maximum Pooling* layers (MaxPooling), *Flatten* layers (Flatten), *Dense* layers (Dense), and *Softmax* layers (Softmax). There was no *striding* in the ConvLayers. Table 1 gives an overview over the entire architecture and the outputs of each layer.[4] The CNN has $179,424$ trainable parameters. The input is a grey scale image. A region of interest of size 192×192 pixels is cropped from the image's center.[5] This size allowed reasonable performance for images of about 500 dpi resolution. It also allows application to small images, e.g. 300×300 images in FVC2000DB1 [Ma02a].
Rotation estimation is obviously a regression task. However, it has been shown that classification can be superior over regression even on regressions task, e.g. if one uses *Deep Expectation*, which averages estimations over classes to form a better estimation [RTVG15]. The method of Deep Expectation has also already been successfully applied to the domain of fingerprint recognition: It was used for fingerprint orientation field estimation [SSB17].

[4] The model was created and trained in the deep learning framework *Tensorflow* [Ab16].

[5] Cropping the region of interest to the center of the fingerprint sample's foreground might be beneficial. However, this approach is independent from such a foreground detection.

Each *class* in the classification approach represents a distinctive angular offset. The outputs of the Softmax in this CNN can be interpreted as probabilities for the angular offsets. Weighted averaging according to the probabilities yields the final estimation. We have tried regression and classification for the task of estimating a rotational alignment. Our experiments yielded classification including Deep Expectation to be more effective than regression for this task.

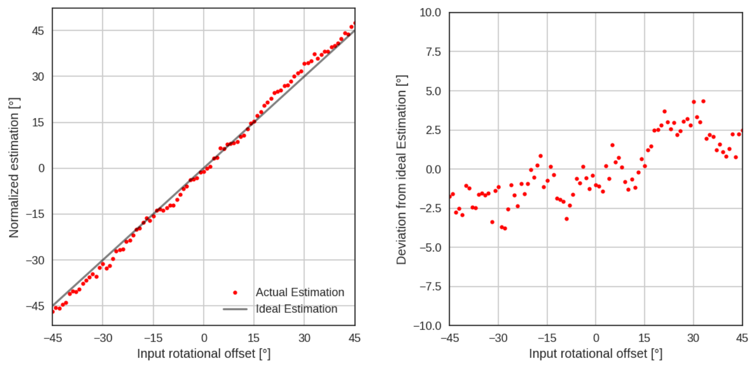
Training Training of the CNN can be roughly summarized as follows: The CNN's task is to learn the rotational difference between two randomly rotated instances of the same fingerprint (see Figure 1).
In detail, during training a fingerprint sample is picked from the training data. Two random values θ_1 and θ_2 for rotations are sampled from a uniform distribution $\mathscr{U}_{[-90^\circ,90^\circ]}$. The fingerprint sample is rotated according to θ_1 and θ_2. The rotated instances are then fed into the CNN, which calculates two estimations $\tilde{\theta}_1$ and $\tilde{\theta}_2$ respectively. We can not make any assumptions about the *initial rotation* of the original fingerprint sample. Thus, θ_1 and θ_2 must be assumed to be biased by this initial rotation. Therefore, minimization of the differences $|\theta_1 - \tilde{\theta}_1|$ and $|\theta_2 - \tilde{\theta}_2|$ likewise does not seem reasonable. To circumvent the lack of knowledge on the initial rotation, we formulate the loss function using the difference between the random rotations and the difference between the estimations:

$$\operatorname{loss}\left(\theta_1, \theta_2, \tilde{\theta}_1, \tilde{\theta}_2\right) = \left|(\theta_1 - \theta_2) - (\tilde{\theta}_1 - \tilde{\theta}_2)\right| \tag{1}$$

By doing so, the CNN learns the rotational difference between two rotations. Minimization $|\theta_1 - \tilde{\theta}_1|$ and $|\theta_2 - \tilde{\theta}_2|$ would have a single optimal solution $\{(\theta_1^*, \theta_2^*) : \theta_1^* = \theta_1, \theta_2^* = \theta_2\}$. The proposed loss function allows an infinite set of optimal solutions $\{(\theta_1^*, \theta_2^*) : (\theta_1^* - \theta_1 = \theta_2^* - \theta_2)\}$. A training sample therefore consists of two random rotation angles and the corresponding rotated instances of the same fingerprint sample. By the way, using two different samples would fail, since their initial rotation is unknown.
Eight pairs of samples were processed as a single batch. We used optimizer *AdaGrad* [DHS11]. Other optimizers like *Stochastic Gradient Descent* or *Adam* did not perform significantly different. Learning rate was set to 10^{-5}.
Training was stopped, when the mean absolute deviations of a validation set did not improve any further. This was done to prevent over-fitting to the training data. Training took only a few hours on a GPU[6]. Estimation of the rotation for a single fingerprint sample takes about 4ms on GPU.

Data Augmentation For CNNs holds in general, that the more appropriate training data is available, also the higher is the performance. A common method to increase the amount of training data is *data augmentation*. Rotating the input data according to the random angular distortions already increases the amount of available training data. In addition, we did some additional augmentation by slightly shifting the input image horizontally and

[6] NVIDIA GTX 780



(a) Regression (b) Deviation

Fig. 2: For testing, the CNN estimates rotations for rotated instances of a given fingerprint sample. Normalization allows to ignore the unknown initial offset of a single sample (2a). A linear regression with unit slope represents an ideal estimation. It allows inspection of the actual deviations (2b).

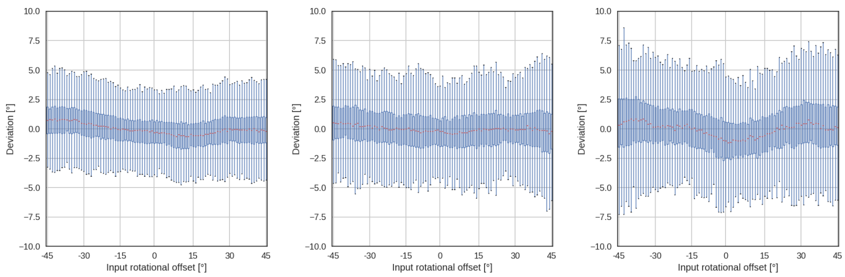
vertically. This is a way to prevent *over-fitting*, i.e. the CNN works significantly better on the training data than on any other data, which has not been seen during training.

4 Experiments

We tested our approach by training the CNN on fingerprint samples from dataset FVC2002 DB1 [Ma02b]. The dataset consists of 100 fingers with eight impressions each. We split the set of 800 fingerprint samples into three parts: the first 400 samples for training, the next 200 samples for validation, and the last 200 samples for testing. By doing so, we took care, that no finger is shared between the sets.
No assumptions on distributions of rotations in the real world can be made. We therefore set up a test scenario, in which each fingerprint sample was rotated by angels from $[-45°, 45°]$. Quantisation step for the angles was $1°$. The trained CNN was then applied to the rotated fingerprints to estimate the rotational offsets. Only differences between rotations are relevant. Plain estimations cannot be used for evaluation. A normalization is necessary. Let μ_θ be the mean of the estimations for all rotated samples of the same fingerprint. We can then calculate a *normalized* estimation $\tilde{\theta}_N(\alpha)$ for estimation $\tilde{\theta}(\alpha)$ for a sample rotated by α by subtracting the mean estimation μ_θ:

$$\tilde{\theta}_N(\alpha) = \tilde{\theta}(\alpha) - \mu_\theta \tag{2}$$

The normalized estimation $\tilde{\theta}_N(\alpha)$ is independent from any initial rotations in the original fingerprint sample. Thus, the normalization yields evaluations in the first place. Then we calculated a linear regression with a unit slope for the normalized estimations for all rotated instances of the fingerprint (see Figure 2). The linear regression represents an ideal estimation. Therefore, the linear regression can be used to evaluate the estimation independent from the unknown initial rotation of the original fingerprint sample.



(a) Training set (b) Validation set (c) Test set

Fig. 3: Visualization of deviations between input orientations and normalized estimation for the entire datasets(see Figure 2 for deviations of a single sample).

Linear regressions are calculated for each fingerprint sample in a dataset. The box plot in Figure 3a visualizes the deviations between estimations and ideal estimations for all samples in the training set. The deviation of the mean for the entire dataset varies about 1° around the ideal estimation. The relative deviation is therefore small compared to the range of the input rotations. The normalized deviation $\tilde{\theta}_N(\mathrm{inp}_\alpha)$ of a given input sample *inp* rotated by angle α can be used to calculate the *mean absolute deviation* δ over M samplings:

$$\delta(\mathrm{inp}) = \frac{1}{M} \sum_{\alpha=-45^\circ}^{45^\circ} \left| \alpha - \tilde{\theta}_N(\mathrm{inp}_\alpha) \right| \tag{3}$$

For a given dataset with N input samples *inp* and a given range of input rotations we can calculate the mean deviation $\bar{\delta}$ per dataset:

$$\bar{\delta} = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathrm{inp}_i) \tag{4}$$

This value $\bar{\delta}$ can be used to estimate the expected deviation on a dataset, i.e. how well the estimation works on a dataset. Using the proposed evaluation method, we evaluated the trained CNN on the training part, the validation part, and the test part of dataset FVC2002DB1. We extended our evaluations also to other datasets to test the CNN for its generalisation capabilities. We therefore chose beyond FVC2002DB1 five additional datasets from the *Fingerprint Verification Contest* (FVC) benchmark series, which also consists of fingerprint samples from optical fingerprint livescanners with a resolution of approx. 500 dpi: FVC2000 DB1 and DB3 [Ma02a], FVC2002 DB2 [Ma02b], and FVC2004 DB1 and DB2 [Ma04]. Only the last 200 fingerprint samples from each dataset were taken for evaluation.
Table 2 summarizes the results, when the CNN is trained on the training set of dataset FVC2002 DB1. Keeping in mind, that the quantisation step for estimation classes was 1°,

Data	Trained on FVC2002 DB1 Mean Abs Dev $\bar{\delta}$	Trained on FVC2000 DB3 Mean Abs Dev $\bar{\delta}$
FVC2002 DB1 Training	1.3°	-
FVC2002 DB1 Validation	1.8°	-
FVC2002 DB1 Test	2.1°	17.2°
FVC2000 DB1	17.2°	35.0°
FVC2000 DB3 Training	-	1.6°
FVC2000 DB3 Validation	-	2.0°
FVC2000 DB3 Test	20.8°	2.6°
FVC2002 DB2	13.9°	40.4°
FVC2004 DB1	6.2°	13.2°
FVC2004 DB2	13.4°	15.4°

Tab. 2: Results when trained on FVC2002 DB1 and FVC2000 DB3.

Fig. 4: Eight samples of a single fingerprint from the test set were rotated according to corresponding estimations by the trained CNN.

the trained CNN achieves a very small mean absolute deviation of only one degree on the training data. While the approach achieves a mean absolute deviation $\bar{\delta}$ of 1.8° on the validation data, 2.1° are achieved on the test data. Figure 4 visualizes eight samples of a single fingerprint. The trained CNN estimated the rotation in the original samples and the samples were rotated according to the estimations.
When applied to other datasets the empirical mean absolute deviations $\bar{\delta}$ range from 6.2° for dataset FVC2004 DB1 to 20.8° for dataset FVC2000 DB3. Fingerprint samples in these datasets have other characteristics than the samples in the training data. A distribution shift in the input data is a typical challenge in pattern recognition. Figure 5 visualizes the cumulative probabilities for absolute deviations between the normalized estimations and the ideal rotations.
As the CNN performed worst on FVC2000 DB3, we wondered whether our approach is applicable to this data at all. If training is successful, the distribution shift can be tackled by training on the relevant data. We therefore trained another CNN for estimations on this database. It achieved a mean absolute deviation of 2.6° on the test part of FCV2000DB3 (see Table 2). It can therefore also be applied to this dataset.

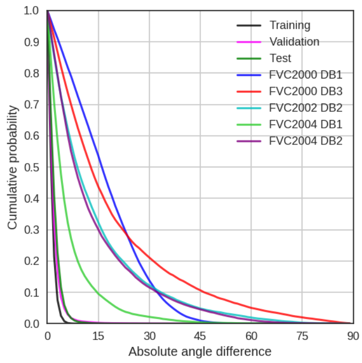


Fig. 5: Cumulative probabilities of exceeding absolute angle differences when trained on dataset FVC2002 DB1.

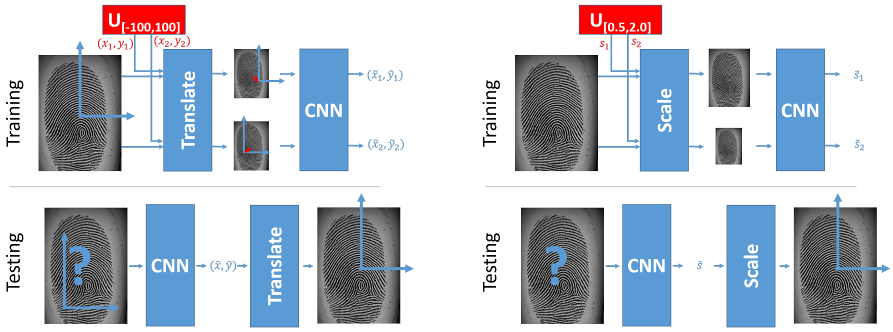
Ref	Approach	Dataset	Performance
[Ho09]	Longitudinal axis	NIST SD4	4.2° mean difference
[LBY14]	Isosceles triangles	CASIA	4.1° mean difference
[Me10]	Ellipse from outline	sequestered	2.3° mean difference
[Ou17]	R-CNN	NIST SD14	95% samples with difference $< 5°$
[YFZ14]	Localized dictionaries	NIST SD27	13.8° mean difference

Tab. 3: Results reported in related work.

Comparisons to other approaches is hardly possible, since there are no other unsupervised approaches for estimation of rotations. Unfortunately, most of the approaches in the related work treat the aspect of rotational alignment as part of a larger workflow and do not explicitly report results for the estimation of the rotation. All reported results from related work can be found in Table 3. Comparison is still difficult, since different metrics and different datasets are used for evaluation. However, the reported results allow some comparison to the proposed approach. Hotz et al. achieved a mean difference of 4.2° on dataset NIST SD4 [WW92]. Li et al. tested their approach on 80 samples with pattern type *Arch* from dataset CASIA. They reported a mean difference of 4.1°. Merkle et al. achieved a mean difference of 2.3° on a sequestered dataset. Ouyang et al. reported a deviation smaller than 5° for 95% of the tested samples of dataset NIST SD14 [WW93]. Yang et al. achieved a mean difference of 13.8° on the latent fingerprint dataset NIST SD27 [GM00].

5 Conclusion

We presented an approach to estimate a rotational alignment for fingerprint samples. The fingerprints are not guaranteed to be upright after alignment. CNNs were trained to estimate an individual rotational offset for each fingerprint. Our main contribution is the definition of a dedicated loss function, which yields an unsupervised training, independent of any initial rotation of the samples. Application of this loss function is not limited to fingerprint samples. It can be applied to any image data.



(a) Translational offset (b) Scaling

Fig. 6: Similar strategies may allow to learn translational offsets (6a) and scaling (6b).

The proposed approach was tested on several datasets containing plain fingerprint samples acquired with livescanners. For data similar to the training data a relative small mean absolute deviation of 2.1° to an ideal alignment can be achieved. The best mean absolute deviation for other datasets is 6.2°. Generalization therefore is an issue. However, if enough appropriate training data is supplied, then the approach can be applied to new data.

6 Future Work

Our approach does only assign an individual rotation to every finger. Extending this approach to train for an absolute rotation alignment may be even more beneficial. In addition, this approach does also tackle only the orientation part of the alignment task. Extending this approach with a translational alignment and scaling would beneficial as well. A similar unsupervised learning strategy for the translational offsets or scaling shall be applicable (see Figure 6). An obvious strategy to further improve the orientation estimation is to apply the CNN to several crops of a single fingerprint sample and apply a voting strategy among the resulting estimations.

References

[Ab16] Abadi, Martín; Barham, Paul; Chen, Jianmin; Chen, Zhifeng; Davis, Andy; Dean, Jeffrey; Devin, Matthieu; Ghemawat, Sanjay; Irving, Geoffrey; Isard, Michael et al.: TensorFlow: A System for Large-Scale Machine Learning. In: OSDI. volume 16, pp. 265–283, 2016.

[Br94] Bromley, Jane; Guyon, Isabelle; LeCun, Yann; Säckinger, Eduard; Shah, Roopak: Signature verification using a" siamese" time delay neural network. In: Advances in Neural Information Processing Systems. pp. 737–744, 1994.

[DHS11] Duchi, John; Hazan, Elad; Singer, Yoram: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.

[GM00] Garris, Michael D; McCabe, R Michael: NIST special database 27: Fingerprint minutiae from latent and matching tenprint images. National Institute of Standards and Technology, Technical Report NISTIR, 6534, 2000.

[Ho09] Hotz, Thomas: Intrinsic coordinates for fingerprints based on their longitudinal axis. In: Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on. IEEE, pp. 500–504, 2009.

[Ja00] Jain, Anil K; Prabhakar, Salil; Hong, Lin; Pankanti, Sharath: Filterbank-based fingerprint matching. IEEE transactions on Image Processing, 9(5):846–859, 2000.

[Ja15] Jaderberg, Max; Simonyan, Karen; Zisserman, Andrew et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025, 2015.

[JM02] Jain, Anil K; Minut, Silviu: Hierarchical kernel fitting for fingerprint classification and alignment. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. volume 2. IEEE, pp. 469–473, 2002.

[Kr14] Krish, Ram P; Fierrez, Julian; Ramos, Daniel; Ortega-Garcia, Javier; Bigun, Josef: Partial fingerprint registration for forensics using minutiae-generated orientation fields. In: Biometrics and Forensics (IWBF), 2014 International Workshop on. IEEE, pp. 1–6, 2014.

[LBY14] Li, Guoqiang; Busch, Christoph; Yang, Bian: A novel approach used for measuring fingerprint orientation of arch fingerprint. In: Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on. IEEE, pp. 1309–1314, 2014.

[Li08] Li, Jianjie; Yang, Xin; Tian, Jie; Shi, Peng; Li, Peng: Topological structure-based alignment for fingerprint fuzzy vault. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, pp. 1–4, 2008.

[LJK05] Liu, Manhua; Jiang, Xudong; Kot, Alex Chichung: Fingerprint reference-point detection. EURASIP Journal on Applied Signal Processing, 2005:498–509, 2005.

[Ma02a] Maio, Dario; Maltoni, Davide; Cappelli, Raffaele; Wayman, James; Jain, Anil: FVC2000: Fingerprint verification competition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(3):402–412, 2002.

[Ma02b] Maio, Dario; Maltoni, Davide; Cappelli, Raffaele; Wayman, James L; Jain, Anil K: FVC2002: Second fingerprint verification competition. In: Pattern recognition, 2002. Proceedings. 16th international conference on. volume 3. IEEE, pp. 811–814, 2002.

[Ma04] Maio, Dario; Maltoni, Davide; Cappelli, Raffaele; Wayman, Jim L; Jain, Anil K: FVC2004: Third fingerprint verification competition. In: Biometric Authentication, pp. 1–7. Springer, 2004.

[Ma09] Maltoni, Davide; Maio, Dario; Jain, Anil K; Prabhakar, Salil: Handbook of fingerprint recognition. springer, 2009.

[Ma18] Markert, Karla; Krehl, Karolin; Gottschlich, Carsten; Huckemann, Stephan F: Detecting Anisotropy in Fingerprint Growth. arXiv preprint arXiv:1801.06437, 2018.

[Me10] Merkle, Johannes; Ihmor, Heinrich; Korte, Ulrike; Niesing, Matthias; Schwaiger, Michael: Performance of the fuzzy vault for multiple fingerprints (extended version). arXiv preprint arXiv:1008.0807, 2010.

[NRV10] Nagar, Abhishek; Rane, Shantanu; Vetro, Anthony: Alignment and bit extraction for secure fingerprint biometrics. In: Media Forensics and Security II. volume 7541. International Society for Optics and Photonics, p. 75410N, 2010.

[Ou17] Ouyang, Jiahong; Feng, Jianjiang; Lu, Jiwen; Guo, Zhenhua; Zhou, Jie: Fingerprint pose estimation based on faster R-CNN. In: Biometrics (IJCB), 2017 IEEE International Joint Conference on. IEEE, pp. 268–276, 2017.

[RA00] Rerkrai, Krisakorn; Areekul, Vutipong: A new reference point for fingerprint recognition. In: Image Processing, 2000. Proceedings. 2000 International Conference on. volume 2. IEEE, pp. 499–502, 2000.

[RTVG15] Rothe, Rasmus; Timofte, Radu; Van Gool, Luc: Dex: Deep expectation of apparent age from a single image. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 10–15, 2015.

[SK14] Sood, Parul; Kaur, Manvjeet: Methods of automatic alignment of fingerprint in fuzzy vault: a review. In: Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, pp. 1–4, 2014.

[SSB17] Schuch, Patrick; Schulz, Simon-Daniel; Busch, Christoph: Deep expectation for estimation of fingerprint orientation fields. In: Biometrics (IJCB), 2017 IEEE International Joint Conference on. IEEE, pp. 185–190, 2017.

[Ta13] Tams, Benjamin: Absolute fingerprint pre-alignment in minutiae-based cryptosystems. In: Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the. IEEE, pp. 1–12, 2013.

[WW92] Watson, Craig I; Wilson, CL: NIST special database 4. Fingerprint Database, National Institute of Standards and Technology, 17:77, 1992.

[WW93] Watson, Craig I; Wilson, CL: Special database 14. S. Department of Commerce, NIST, Advanced Systems Division, Gaithersburg, Maryland. Citeseer, 1993.

[YB09] Yang, Bian; Busch, Christoph: Parameterized geometric alignment for minutiae-based fingerprint template protection. In: Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on. IEEE, pp. 1–6, 2009.

[YFZ14] Yang, Xiao; Feng, Jianjiang; Zhou, Jie: Localized dictionaries based orientation field estimation for latent fingerprints. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 36(5):955–969, 2014.

[ZFH14] Zhang, Xinglong; Feng, Quan; He, Kang: A new blind fingerprint alignment algorithm used in biometric encryption. In: 2014 International Conference on Computer, Communications and Information Technology. 2014.