Estimating the Data Origin of Fingerprint Samples

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

Abstract: The data origin (i.e. acquisition technique and acquisition mode) can have a significant impact on the appearance and characteristics of a fingerprint sample. This dataset bias might be challenging for processes like biometric feature extraction. Much effort can be put into data normalization or into processes able to deal with almost any input data. The performance of the former might suffer from this general applicability. The latter losses information by definition. If one is able to reliably identify the data origin of fingerprints, one will be able to dispatch the samples to specialized processes. Six methods of classification are evaluated for their capabilities to distinguish between fifteen different datasets. Acquisition technique and acquisition mode can be classified very accurately. Also, most of the datasets can be distinguished reliably.

Keywords: fingerprint recognition, machine learning, dataset bias.

1 Introduction

No two fingerprints are the same. Every fingerprint is at least slightly different from another. This fact makes a fingerprint a valuable trait for biometric recognition. However, two fingerprint samples of the very same fingerprint can also be very different. An important source of variation arises from the two aspects of a sample's *data origin*: *acquisition technique* and *acquisition mode*.

Fingerprint samples might have been acquired with different techniques. For example, they can be acquired using dedicated fingerprint livescanners. There is a variety of manufacturers and devices. The latter can differ in physical acquisition principles, e.g. optical or capacitive. Besides dedicated devices, almost any camera may be used for acquisition. Fingerprints may also be acquired using *ink* and paper. There are also *latent* fingerprints or *fingermarks*, which are typically evidences from crime-scenes. Those fingerprints are captured using special techniques, e.g. photography of fingerprints highlighted with powders.

Not only the technique of acquisition matters, but also *how* the finger is presented. There are namely four modes of acquisition. The finger may be placed *plain* on an acquisition surface. Fingerprints can be *rolled* over an acquisition surface. Some devices require the fingerprint to be *swiped* over a line scanner. While these three modes are all contact-based, the fingerprint may also be acquired *contact-less*. Many of the possible combinations of technique and mode are actually deployed in operational scenarios..

In general, the *fidelity* of a fingerprint sample to its source depends on the data origin [IS16]. Thus, all data is *biased* by its origin. This *dataset bias* tends to be very different among different datasets. Dealing with such differences can be challenging for any process

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

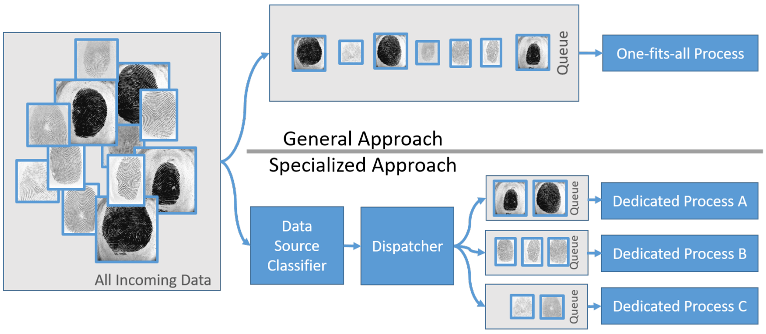


Fig. 1: Knowing a sample's origin enables usage of specialized processing methods.

using the data. And it turns out that biometric comparison across different data origins is challenging [Ji12].

There are two ways of dealing with the dataset bias (see Figure 1). *One-fits-all solutions* need to be able to deal with any input at hand. This general applicability comes at the cost of recognition performance. *Dedicated* or *specialized modules* modules can be tailored to the special needs of an input. The more the inputs differ, the larger is the benefit of a specialized processing pipeline. If you do not know, what you are processing, you will have to apply a one-fits-all solutions. But if you do know, what kind of data you are processing, you will be able to benefit from specialization. There are standardized data interchange formats, which provide meta information about acquisition mode and technique, e.g. the international standard ISO/IEC 19794-4 [IS11]. But fingerprint samples do not always come with such information about their origin or the information is not reliable. This lack of information may be unintentional, e.g. when processing legacy data in a system, in which the samples have various data origins. Or it may be by intention, e.g. in benchmarks or competitions like FVC-ongoing [Do09]. If you still want to apply specialized approaches, you will have to guess the origin of the input data. Guessing the origin essentially makes use of dataset bias, since this bias is what makes data distinguishable. In this case, the dataset bias is a desirable property.

Ghiani et al. proposed linear *Support Vector Machines* for pairwise discrimination of datasets [Gh17]. This is the first extensive investigation of methods for estimating the origin of fingerprint samples using a multitude of datasets. In this work, we propose to use a *Convolutional Neural Network* (CNN) for guessing the origin of the fingerprint. To assess this approach, its performance is then compared to other prominent classifiers.

The rest of this work is organized as follows: Section 2 gives an overview on related research. A CNN based approach is described in Section 3. Section 4 describes the experiments with this approach and five alternative classifiers. It also presents the results achieved. A conclusion can be found in Section 5.

2 Related work

Capabilities of generalization are always important for methods of pattern recognition. If a pattern recognition method is not able to deal with other data than the data used for training, the method will be useless. The dataset bias, also known as the *covariate shift*, might be the most challenging aspect for generalization. Dealing with the dataset bias is therefore an important topic in pattern recognition. A lot of research has been done in enabling methods to deal with data, that has not been seen during the training phase.
Besides the improvements in the generalization capabilities, there is another way to look at the dataset bias. For example, Torralba and Efros investigated how well image datasets can be distinguished [TE11]. They evaluated their method on datasets available at the time for large classification benchmarks. They called this task the *Name the Dataset* game. Their motivation was to measure and to understand the bias in datasets. This bias may usually result in generalization problems for classifiers, when the test data differs from the training data.
The challenge of generalization becomes even more important as the accuracies rise. In case of high accuracies, slight differences in performances between training and test data can result in significant relative differences between these performances. CNNs are state of the art in several domains of image processing and pattern recognition. The accuracies achieved with CNNs are remarkably high and dataset biases can have a strong impact here. Tommasi et al. did extensive experiments on the dataset bias when using CNNs [To17].
The dataset bias is a prominent challenge in fingerprint recognition. This challenge is also known as *cross device biometric recognition*. Jia et al. developed a dataset, which contains fingerprint samples from the same fingerprint acquired with nine different devices [Ji12]. They showed that recognition across different devices is challenging.
Ghiani et al. proposed to use a linear *Support Vector Machine* for classification of different datasets [Gh17]. They used the classification to apply specialized methods of presentation attack detection on the fingerprint samples. They evaluated classification between pairs of datasets. This work provides comparison of six methods of classification and evaluation on a multitude of fingerprint datasets.

3 CNNs for Data Origin Estimation

Estimating the data origin is a typical *classification* task. CNNs are currently state of the art in classification tasks. We therefore propose to use a CNN for this task.
In the following we will sketch an architecture for a CNN, which is capable of this task. We will also provide information on how the training of the CNN was performed.

Architecture Table 1 gives an overview over the entire architecture and the outputs of each layer. There has been no extensive optimization of any hyperparameters. The entire model is built from six different types of layers: *Convolutional* layers (ConvLayer), *Parametric Rectified Linear Units* (PReLu), *Maximum Pooling* layers (MaxPooling), *Flatten* layers (Flatten), *Dense* layers (Dense), and *Softmax* layers (Softmax). All ConvLayer have

Tab. 1: Straight forward architecture of proposed CNN.

#	Layer	Output
0	Input (192×192×1)	(192×192×1)
1	ConvLayer (32×3×3×1)	(190×190×32)
2	MaxPooling (2×2)	(95×95×32)
3	PReLU	(95×95×32)
4	ConvLayer (32×3×3×32)	(93×93×32)
5	MaxPooling (2×2)	(46×46×32)
6	PReLU	(46×46×32)
7	ConvLayer (32×3×3×32)	(44×44×32)
8	MaxPooling (2×2)	(22×22×32)
9	PReLU	(22×22×32)
10	ConvLayer (32×3×3×32)	(20×20×32)
11	MaxPooling (2×2)	(10×10×32)
12	PReLU	(10×10×32)
13	ConvLayer (32×3×3×32)	(8×8×32)
14	MaxPooling (2×2)	(4×4×32)
15	PReLU	(4×4×32)
16	Flatten	(512)
17	Dense (32×512)	(32)
18	PReLU	(32)
19	Dense (32×32)	(32)
20	PReLU	(32)
21	Dense (15×)32	(15)
22	Softmax	(15)

32 kernels. All kernels are 3x3 filters. There was no *striding* in the ConvLayers. MaxPooling was performed on 2x2 blocks. The CNN has less than 60,000 trainable parameters.
Generalization is always a crucial issue when training a classifier. In this approach, the capability of generalization was enforced by adding a l_2-regularization on the kernels of the Dense layers. A common approach to strengthen the generalization capabilities of a CNN is the introduction of *Batch Normalization* [IS15]. However, usage of Batch Normalization in this approach led to an undesired behavior: The CNN was not able to distinguish between the different datasets any more.
The input to the CNN is a gray scale image, which is cropped to its central region of size 192x192 pixels (see Figure 2). The approach does not rely on any foreground detection, it simply crops a region of interest from the center. This allows an automatic processing. The CNN therefore does only see a small part of the fingerprint sample.
The output of the Softmax layer can be understood as the likelihood for the fifteen respective classes of datasets (see Section 4). The model was created and trained in the deep learning framework *Tensorflow* [Ab16].

Tab. 2: Datasets used for evaluation.

#	Dataset	Acquisition Technique	Acquisition Mode	Ref
1	FVC2000 DB1	Optical	Plain	[Ma02a]
2	FVC2000 DB2	Capacitive	Plain	
3	FVC2000 DB3	Optical	Plain	
4	FVC2002 DB1	Optical	Plain	[Ma02b]
5	FVC2002 DB2	Optical	Plain	
6	FVC2002 DB3	Capacitive	Plain	
7	FVC2004 DB1	Optical	Plain	[Ma04]
8	FVC2004 DB2	Optical	Plain	
9	FVC2004 DB3	Thermal	Swiped	
10	FVC2006 DB2	Optical	Plain	[Ca07]
11	FVC2006 DB3	Thermal	Swiped	
12	MCYT DP	Optical	Plain	[Or03]
13	MCYT PB	Capacitive	Plain	
14	NIST DB4	Ink-based	Rolled	[WW92]
15	NIST SD14	Ink-based	Rolled	[Wa01]

Training The model was trained with the optimizer *Adam* [KB14] . There were no significant differences when using optimizers *Adagrad* or *SGD*. Learning rate was set to $\lambda = 10^{-4}$.
The number of samples per batch was selected to be 128. Larger batch sizes did not improve performance. Smaller batch sizes resulted in instabilities during training. The network was trained to minimize cross validation loss. The samples in each batch were randomly picked from the training data. *Data augmentation* was applied to the fingerprint samples. The samples were rotated randomly. The 192x192 pixel region used as the input was cropped randomly from a region nearby the sample's center. Such a data augmentation is commonly used to increase the amount of training data. This indirectly prevents the CNN from overfitting to the training data. Therefore it also helps enabling generalization of the CNN.
For training the CNN, the training set was split into two parts. About two thirds of the fingerprint samples were used as training data. The remaining fingerprint samples were used as a validation set. Training was stopped when the improvement of loss for the validation set stopped. Such an *early stopping* strategy is a common method to prevent a CNN from over-fiting. Training the CNN took less than an hour on a GPU[4].

4 Experiments

Datasets The fifteen tested datasets are a sub set of publicly available datasets. Only publicly available dataset were chosen to allow reproducibility of the results. Even though,

[4] NVIDIA GTX 780

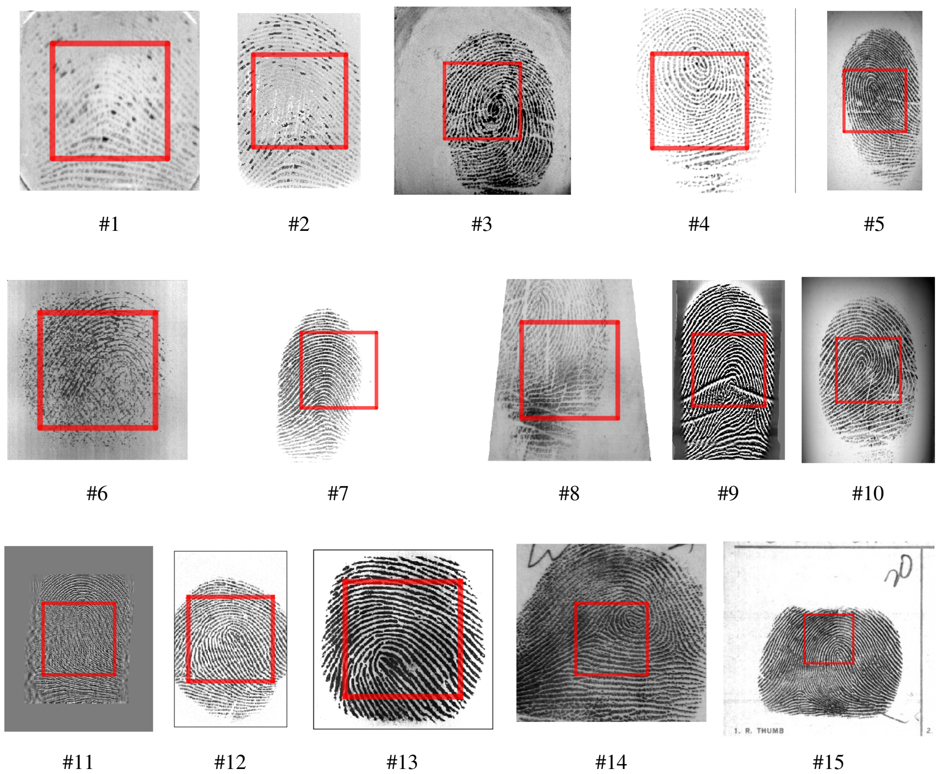


Fig. 2: Samples from the datasets used for evaluation (see Table 2 for numbering). The red square indicates the 192x192 crop region used for training and testing.

some dataset might be some kind of out-dated, they still represent legacy data. Each dataset represents its own class for classification. Table 2 summarizes details on the single dataset used during our experiments. These datasets represent a subset of the variability of *acquisition techniques* and *acquisition modes*. [5] There are datasets, which were acquired by livescanners using optical, capacitive, and thermal sensors. There are also two datasets acquired by using ink-based techniques. Most of the datasets contain plain fingerprints. Two datasets contain rolled fingerprints and two datasets contain swiped fingerprints. Each data will represent one class in the classification. Thus, it will be a multi-class classification.

The number of samples in the datasets differs. Using all samples in the evaluation would have imbalanced the influence of each dataset. Therefore, only the first 800 samples in each dataset have been selected for these experiments. By doing so, all datasets have the same amount of training data and testing samples respectively.
The images were cropped to their central region of 192x192 pixels. This prevents the clas-

[5] In terms of ISO/IEC 19794-4 the acquisition technique may be deducted from the *capture device ID* identifier. The acquisition mode is represented by the *impression type* identifier in the standard.

sifiers to learn from trivial features like image dimensions or any systematic artifacts at the borders of the fingerprint samples.

Metrics The performance of the classifiers was assessed by the *accuracy* acc in predicting the correct data origin. Three accuracies were evaluated. First, the accuracy $\text{acc}_{\text{dataset}}$ of estimating the correct dataset was evaluated. Second, the accuracy acc_{mode} for estimating the correct acquisition mode was measured. Third, the accuracies acc_{tech} for estimating the correct acquisition technique was measured.
Let a tuple (x_i, y_i) contain the i-th fingerprint sample x_i and its actual class $y_i \in Y$ in the set Y of all classes. Let $F(x_i)$ be the estimated class for fingerprint sample x_i. The accuracy $\text{acc}_{\text{dataset}}$ for a set $X = \{x_i : i \in [1, N]\}$ containing N samples is therefore the expected value for the rate of correct estimations for the dataset and can be calculated by using the indicator function $\mathbb{1}$:

$$\text{acc}_{\text{dataset}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F(x_i)=y_i} \tag{1}$$

Let $F_t(y) = m : y \in Y \mapsto t \in T$ be the function that maps a class y to it corresponding mode t in the set T of all acquisition modes, i.e. $T = \{$'Optical', 'Capacitive', 'Thermal', 'Ink-based'$\}$. Then the accuracy acc_{tech} can be calculated as follows:

$$\text{acc}_{\text{tech}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F_t(F(x_i))=F_t(y_i)} \tag{2}$$

This accuracy acc_{tech} can be understood as the expected value for the rate of correct estimations for the acquisition technique.
Respectively, let $F_m(y) = m : y \in Y \mapsto m \in M$ be the function that maps a class y to it corresponding mode m in the set M of all acquisition modes, i.e. $M = \{\text{'Plain'}, \text{'Rolled'}, \text{'Swiped'}\}$. Then the accuracy acc_{mode} can be calculated as follows:

$$\text{acc}_{\text{mode}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{F_m(F(x_i))=F_m(y_i)} \tag{3}$$

This accuracy acc_{mode} can be understood as the expected value for the rate of correct estimations for the acquisition mode.
The priors of the acquisition modes and acquisition technique are not equally distributed over all classes. This imbalance has of course impact on the respective accuracy measures.
4-fold cross-validation was used to allow a more reliable evaluation. Each fingerprint dataset was therefore split into four parts of equal size. In each fold of the evaluation, one of the parts was kept out of the training data and used for testing only. No fingerprint sample is in more than one testing split. The splits were performed randomly. All datasets have more than one fingerprint sample per fingerprint. It was enforced, that all samples

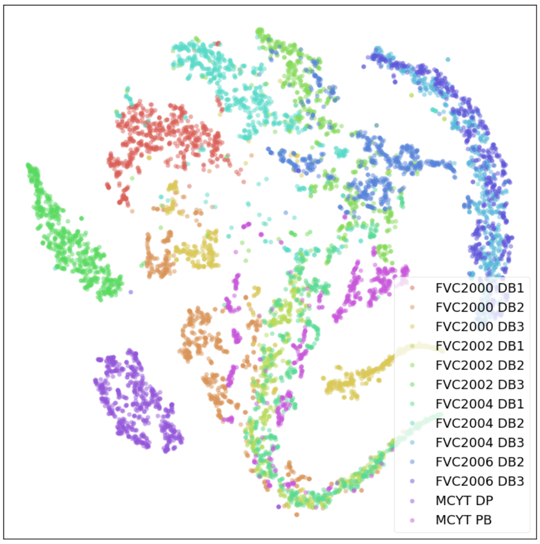


Fig. 3: t-SNE embedding for GLCM features already allows distinguishing between some datasets.

of the same fingerprint are in the same split. Therefore no two samples belonging to the same finger are in the training split and the test split of the same fold. The fact, that fingerprint samples stemming from the same source (finger instance) were splitted into different datasets, was neglected. The accuracies reported here are actually the mean accuracies over all four evaluation runs.
For inspection of the failures in classifying the datasets, *confusion matrices* are calculated. Those matrices allow to analyze the failures made with respect to the real data origin.

Alternative Classifiers Five alternative classifiers were tested to benchmark the performance of the CNN based approach: Random Forest Classifier [Br01], Extra Trees Classifier [GEW06], Decision Tree Classifier, Logistic Regression and K Nearest Neighbor Classifier.[6] All alternatives have implementations in the python based machine learning toolbox *scikit-learn* and can be used out of the box [Pe11].
Classification applied directly to the signal of the central crops did not perform well. Thus, *Gray level co-occurrence matrices* (GLCM) have been calculated for each central crop. [7] Those represent the entire range of gray level values in a crop and the dynamics of neighboring pixels [Ha73]. The intensities of gray values were subsampled by a factor of 4 to reduce the number of features to a reasonable order.
A common step in classic pattern recognition is to do *Feature Selection*. *Principal Component Analysis* (PCA) is probably the most standard method here. PCA analyzes input data for their components of maximal variance. PCA transforms input data to a new base.

[6] AdaBoost, Huber Regressor, and SVM were also tested but failed totally to learn a classification.
[7] GLCM is a classic feature for texture classification. Other features for texture classification may be applied, of course. However, GLCM already yielded impressive results for data origin classification.

Tab. 3: The most relevant accuracies are those achieved on test data. Best result is marked in bold.

Input	Classifier	Feature Selection	$acc_{dataset}$ Train	$acc_{dataset}$ Test	acc_{mode} Train	acc_{mode} Test	acc_{tech} Train	acc_{tech} Test
GLCM	Decision Tree	none	100%	89.2%	100%	97.4%	100%	96.7%
		PCA	100%	83.2%	100%	93.5%	100%	91.6%
	Extra Trees	none	100%	92.9%	100%	98.5%	100%	98.3%
		PCA	100%	90.2%	100%	97.0%	100%	96.2%
	Random Forest	none	99.8%	93.2%	100%	98.6%	100%	98.4%
		PCA	99.8%	91.1%	99.9%	97.0%	99.9%	96.4%
	Logistic Regression	none	100%	**95.6%**	100%	98.1%	100%	97.9%
		PCA	84.3%	83.3%	94.9%	94.5%	93.6%	93.0%
	K Nearest Neighbors	none	92.8%	86.7%	98.1%	96.6%	97.4%	95.2%
		PCA	92.6%	86.4%	98.0%	96.5%	97.3%	95.1%
Images	CNN	none	100%	94.7%	100%	**99.7%**	100%	**99.5%**

This base allows to select only those axes with the largest variance in the data. Figure 3 visualizes a two dimensional embedding of the reduced feature set, which was generated by *t-distributed stochastic neighbor embedding* (t-SNE) [MH08]. Obviously, the input features allow distinguishing between most datasets.

All approaches were evaluated on the full set of input features and also on a reduced feature set. The reduced set contained the important components, which together explain more than 99.9% of the variance in the original data. The PCA based transformation was calculated on the training data only, of course.

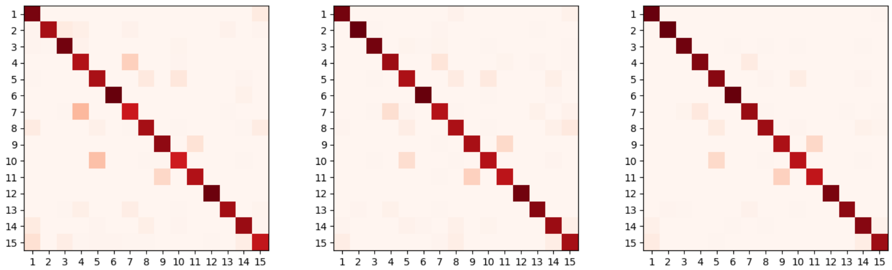
Results Table 3 summarizes the results of the evaluated methods. Figure 4 visualizes the confusion matrices for classifications based on the features derived from entire GLCMs.

Most of the classifiers were able to classify the acquisition mode and the acquisition technique very reliable. The CNN based approach achieved an accuracy of 99.7% for estimating mode and 99.5% for technique respectively. The best alternative classifier achieved similar accuracies.

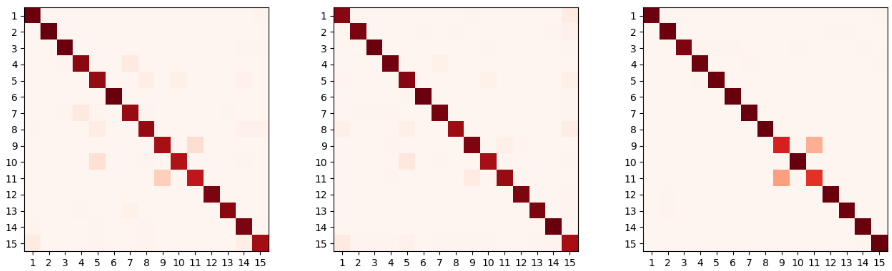
Despite this, dataset classification is more challenging. Most of the classifiers are able to distinguish very reliably even between the different datasets. Of course, the most important aspect is the accuracy achieved for the test sets. The best result of all evaluated approaches is achieved by the Logistic Regression: 95.6% of the samples were classified correctly.

Some pairs of datasets were confused more often than others. Failures are made by confusing samples from the two thermal/swipe datasets. Samples from both datasets containing rolled fingerprints were also confused in some cases. Finally, samples from datasets FVC2002D DB2 and FVC2006 DB2 often were also misclassified. Actually, both datasets were acquired with scanners from the same manufacturer. It is likely, that the same scanner model or even the very same scanner was used for acquisition. If so, the classifiers based their decision on additional dataset biases, e.g. environmental influences.

All classifiers overfitted to the training data. All training data could be classified almost perfectly by all classifiers. However, using the PCA for feature selection did not work out



(a) K Nearest Neighbors (b) Decision Tree (c) Extra Trees



(d) Random Forest (e) Logistic Regression (f) CNN

Fig. 4: The confusion matrices can be used to identify class-specific failures made by a classifier. Each dataset is identified by the numbering provided in Table 2. Most failures in classification are made between datasets 9 and 11 (both thermal swipe), between datasets 15 and 16 (both ink-based rolled), and datasets 5 and 10 (both likely the same scanner).

well. In general, the accuracy achieved on the test data was lower, when feature selection was applied before classification.

5 Conclusion

The dataset bias can be a challenge for any process, which has to deal with unknown input data. We propose to exploit the database bias. If one can use the dataset bias as a distinguishing property for the origin of a fingerprint sample, one will be able to use this information to dispatch the sample to a process, which is specialized on such inputs. Five classifiers to guess the origin of a fingerprint were evaluated. Acquisition mode and acquisition technique were classified very reliable. Fifteen datasets containing their individual dataset biases were tested for evaluation. Most of the conventional classifiers worked well out of the box: Accuracy for the estimation was over 95%. The classification errors do not distribute equally among the different classes. While most of the datasets were distinguished reliably by the classifiers, some are harder to be distinguished. The CNN based approach and the conventional approaches performed similar.

Acknowledgments

The authors thank the *15th IAPR/Eurasip International Summer School for Advanced Studies on Biometrics for Secure Authentication* for inspiration and all suppliers of datasets used in our experiments.

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.

[Br01] Breiman, Leo: Random forests. Machine learning, 45(1):5–32, 2001.

[Ca07] Cappelli, Raffaele; Ferrara, Matteo; Franco, Annalisa; Maltoni, Davide: Fingerprint verification competition 2006. Biometric Technology Today, 15(7):7–9, 2007.

[Do09] Dorizzi, Bernadette; Cappelli, Raffaele; Ferrara, Matteo; Maio, Dario; Maltoni, Davide; Houmani, Nesma; Garcia-Salicetti, Sonia; Mayoue, Aurélien: Fingerprint and on-line signature verification competitions at ICB 2009. In: International Conference on Biometrics. Springer, pp. 725–732, 2009.

[GEW06] Geurts, Pierre; Ernst, Damien; Wehenkel, Louis: Extremely randomized trees. Machine learning, 63(1):3–42, 2006.

[Gh17] Ghiani, Luca; Mura, Valerio; Tuveri, Pierluigi; Marcialis, Gian Luca: On the Interoperability of Capture Devices in Fingerprint Presentation Attacks Detection. In: ITASEC. pp. 66–75, 2017.

[Ha73] Haralick, Robert M; Shanmugam, K; Dinstein, ItsâHak et al.: Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, 3(6):610–621, 1973.

[IS11] ISO/IEC JTC1 SC37 Biometrics: ISO/IEC 19794-4:2011 Information technology - Biometric data interchange formats - Part 4: Finger image data. Technical report, International Organization for Standardization, 2011.

[IS15] Ioffe, Sergey; Szegedy, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[IS16] ISO/IEC JTC1 SC37 Biometrics: ISO/IEC 29794-1:2016 Information technology - Biometric sample quality - Part 1: Framework. ISO 29794-1, International Organization for Standardization, Geneva, Switzerland, 2016.

[Ji12] Jia, Xiaofei; Yang, Xin; Zang, Yali; Zhang, Ning; Tian, Jie: A cross-device matching fingerprint database from multi-type sensors. In: Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, pp. 3001–3004, 2012.

[KB14] Kingma, Diederik P; Ba, Jimmy: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[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.

[MH08] Maaten, Laurens van der; Hinton, Geoffrey: Visualizing data using t-SNE. Journal of machine learning research, 9(Nov):2579–2605, 2008.

[Or03] Ortega-Garcia, Javier; Fierrez-Aguilar, J; Simon, D; Gonzalez, J; Faundez-Zanuy, M; Espinosa, V; Satue, A; Hernaez, I; Igarza, J-J; Vivaracho, C et al.: MCYT baseline corpus: a bimodal biometric database. volume 150. IET, pp. 395–401, 2003.

[Pe11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[TE11] Torralba, Antonio; Efros, Alexei A: Unbiased look at dataset bias. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 1521–1528, 2011.

[To17] Tommasi, Tatiana; Patricia, Novi; Caputo, Barbara; Tuytelaars, Tinne: A deeper look at dataset bias. In: Domain Adaptation in Computer Vision Applications, pp. 37–55. Springer, 2017.

[Wa01] Watson, Craig I: NIST Special Database 14: Mated Fingerprint Cards Pairs 2 Version 2. Technical report, Citeseer, 2001.

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