Soft Biometrics Classification Using Denoising Convolutional Autoencoders and Support Vector Machines

Nelson Marcelo Romero Aquino[1], Matheus Gutoski[2]
Leandro Takeshi Hattori[3] and Heitor Silvério Lopes[4]

Federal University of Technology - Paraná
Av. Sete de Setembro, 3165 - Rebouças CEP 80230-901
[1] nmarceloromero@gmail.com
[2] matheusgutoski@gmail.com
[3] lthattori@gmail.com
[4] hslopes@utfpr.edu.br

Abstract. This work presents a methodology to perform the classification of soft biometrics in images of pedestrians using a Denoising Convolutional Autoencoder as feature extractor and a Support Vector Machine as classifier. The Denoising Convolutional Autoencoder was trained with a custom dataset containing a combination of five available datasets (3DPES, Market1501, PRID2011, VIPeR and ETHZ) and used as a feature extractor of the images of the VIPeR dataset. The extracted features were then used as input values for a Support Vector Machine classifier, with its hyper-parameters set by using Grid Search, in order to classify the images according to two soft biometrics or labels: Long-Hair and Sunglasses. The results obtained with the proposed approach were compared to those obtained using other well-known feature extractor: Histogram of Oriented Gradients.

Keywords: Deep Learning, Convolutional Neural Networks, Data Augmentation, Unbalanced Datasets

1 Introduction

Soft biometrics are human characteristics, physiological or behavioral, which provide information that allows to differentiate two individuals. Although soft biometrics are usually not unique for each individual, they can provide some prior information about the subjects. Furthermore, using just one soft biometric may not be a suitable option to identify individuals, however, a combination of them can lead to satisfactory results. Soft biometrics can also be used to complement other primary biometric identifiers such as fingerprints and faces. Some common soft biometrics are: gender, age, height, weight, clothes color, tattoos and hair length.

Over the years, the necessity to increase public security produced the growth of the number of surveillance cameras installed in public places. They allow

to obtain images and videos in real time without much effort. Hence, a lot of information can be extracted from these resources to solve different types of problems. The identification of individuals in the images obtained by surveillance cameras is one of those problems. For this task, soft biometrics are useful, since they provide information that can be used to differentiate the subjects in the images. Nonetheless, this is an exhaustive process to be done by a human observer. Therefore, computer vision plays an important role in this problem.

Several works aimed at solving this problem during the recent years. Considering the Deep Learning (DL) [1] approaches, one of the most used approaches is based on the use of Convolutional Neural Networks (CNNs) [2]. In [3], two CNNs for classifying three soft biometrics traits (Upper Clothes, Lower Clothes and Gender) were presented. Whilst [4] estimated the age and gender of individuals from human faces in images using a CNN. Works by [5] and [6] presented architectures based on training the feature extraction part of a CNN (Convolution and Pooling layers) separately for each patch of an image and joining the features into a flattened vector that is fed to the Fully Connected layers of the network. Both approaches allow to perform multi-label classification. All works used variations of the Stochastic Gradient Descent (SGD) method [7].

DL approaches are also trained to serve as feature extractors. For instance, Wang [8] presented an approach based on a 6-layer architecture CNN to serve as supervised feature extractor in order to also estimate age from images containing faces. On the other hand, [9] proposes the use of Stacked Convolutional Autoencoders (SCA) for unsupervised feature learning to initialize a CNN with filters of the trained SCA. Following this line, this work presents a method to extract features to perform the classification of soft biometrics of pedestrians in images. A Denoising Convolutional Autoencoder (DCA) is used as feature extractor and Support Vector Machine (SVM) is used as classifier. The results obtained when classifying samples from the VIPeR Dataset [10] using the DCA were compared to those obtained when using Histogram of Oriented Gradients (HOG) [11].

This work is organized as follows: Section 2 presents the theoretical aspects of the methods. Section 3 presents the methodology, Section 4 presents the datasets used in this work. Section 5 presents the experiments and results, and Section 6 presents the conclusions and points directions for future works.

2 Background

Image classification is the process of separating images according to their content after a pre-processing phase and the extraction of their features. For the particular problem of soft biometric traits in images of individuals, the set of images can be classified, for instance, by those that contain people with long hair and those that do not. Traditionally, image classification consists of the following steps: image segmentation, feature extraction, feature selection, classification and validation.

2.1 Autoencoders

Autoencoders (AE), also known as Autoassociators or Diabolo Networks [12], are neural networks capable of generating a latent representation of its input matrix and giving as output a reconstruction of that original input. Usually, AE are used to obtain a representation of the original data with a reduced dimensionality. The training process of the AE is unsupervised, since labels are not necessary. When an AE is trained, the objective is to minimize some measure of dissimilarity, usually the Mean Squared Error (MSE) [13], considering the original input of the network and its output. Thus, the number of input units of the network must be equal to the number of output units. An AE has two main parts, the Encoder, which is a single or a group of layers that allow to reduce the dimension of the original data, and the Decoder, which performs the opposite operation to produce the approximate reconstruction of the original input. Figure 1 presents an AE with a simple architecture based on three fully-connected hidden layers.

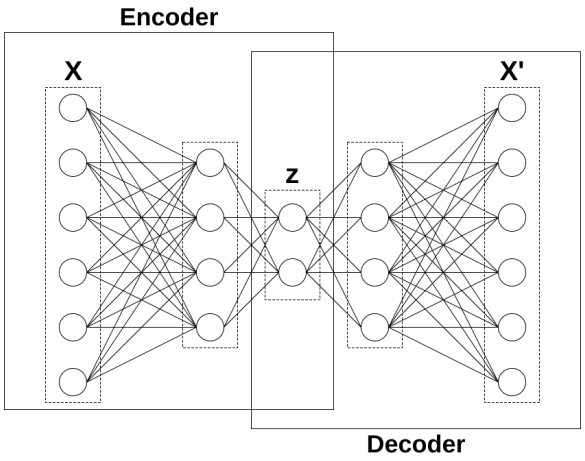

Fig. 1: An Autoencoder with three fully-connected hidden layers. X is the input layer, $X´$ the output layer, which is the approximate reconstruction of X. The output of the second hidden layer is the latent representation z.

There are several methods to improve the performance of AE. A common option is to use partially corrupted inputs during the training process, so that the network can improve its capability to represent the original input. Networks trained with this technique are called Denoising Autoencoders (DCA) [14].

For problems related to images, an alternative to the original AE is the Convolutional Autoencoder, which is based on stacking convolutional layers [15]. This could be useful due to that convolutional layers allow the network to learn high and low level features of images. The Decoder has an inverted structure, having fully-connected layers at the beginning and convolutional layers at the end. For the decoding, a special type of convolution is used: deconvolution [16], which reverts the effects of the convolution.

The AE used in this work was composed of convolution layers only and online corrupted input images with Gaussian noise were used during the training phase. The final features are extracted by flattening the last layer of the encoder.

3 Methodology

The objective of this work is to classify soft biometrics images by using a Denoising Autoencoders (DCA) as a feature extractor and a SVM as the classifier. This section presents the details and parameters of the methods used in this work. Figure 2 briefly shows each stage of the methodology.

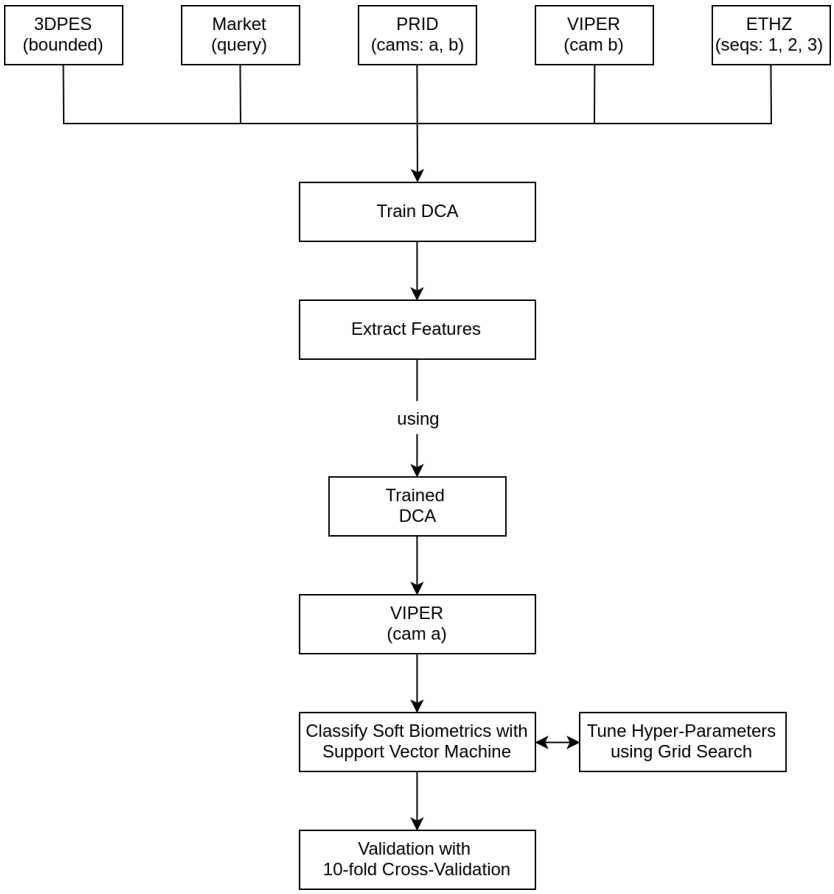

Fig. 2: Phases of the soft biometrics classification process.

3.1 Denoising Convolutional Autoencoder

The AE is Denoising, that is, during the training its inputs were partially corrupted with Gaussian noise and propagated through the network to obtain the output, which is then compared to the original image (without the corruption)

using MSE as loss function. The MSE metric is presented at Equation 1, where X is the original image and $X´$ is the reconstruction for n training samples.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(X - X')_j^2 \quad (1)$$

The weights of the network were updated using the Adaptive Moment Estimation (Adam) algorithm [17] searching to minimize the loss function. Adam is a stochastic optimization method which only requires first-order gradients and has little memory requirement. The method is based on the computation of adaptive learning rates for each parameter from estimates of first and second moments of the gradients.

The network is also a fully Convolutional AE since all of its hidden layers are convolutional or deconvolutional. The architecture, with a fully convolutional structure that is partially inspired on the fully Convolutional AE presented at [15], is shown at Figure 3. It contains twelve hidden layers, the first six are convolutional layers (Encoder) and the following six are deconvolutional layers (Decoder). The output of the last convolutional layer, *Conv6* at Figure 3, is the latent representation z (a vector with 256 dimensions) of the input X, which is a 128x128 image with three channels (Red, Green and Blue). z is then propagated through the deconvolutional layers to produce the output $X´$, that is, the reconstruction of X.

The weights of the hidden layers are initialized with a truncated normal distribution with standard deviation equal to 0.1. A filter size of 3x3 is used for all layers and Rectified Linear Units (ReLU) are used as non-linearity [18, 1]. All biases are initialized with zeros. Since there has not been used any type of pooling to downsample the outputs of the convolutional layers, the downsample was done by using strides equal to 2x2 for the convolutions, except for the last convolutional layer *Conv6* and the first deconvolutional layer *Deconv1*, both of which use 1x1 strides.

The network is configured and trained using the TensorFlow framework [19]. The network is trained for 100 epochs and uses a learning rate equal to 0.0001. The L2 Regularization method was also applied in order to reduce overfitting. This method, adds an extra term to the cost function in order to force the network to learn preferably small weights. The regularization parameter or weight decay λ was set to 0.0005.

A NVIDIA Titan X Pascal GPU was used to train the network, built within a computer with 12 CPUs with 1.2GHz clock rate and 32Gb RAM.

3.2 HOG Configuration

HOG [11] was used to extract features from the images using the Python library Scikit-Image [20]. The features were extracted from local regions with 16 x 16 pixels and the histograms of edge intensity were calculated from 2 x 2 local cells with 8 orientations. With this configuration, a HOG feature vector of 3872 dimensions is obtained for each image of the VIPeR (cam a) dataset.

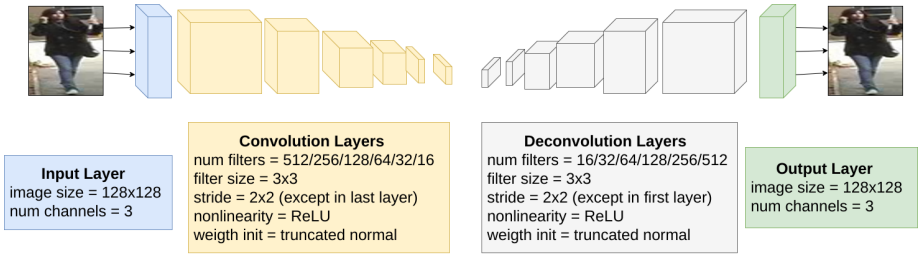

Fig. 3: The architecture of the Denoising Convolutional Autoencoder. The bottom part presents the shapes taken by the data during the propagation through the network after each convolution or deconvolution, and some basic informations about the layers.

3.3 Training and Validating the SVM models

The SVM models are trained using a radial basis function kernel and grid search method to automatically set the hyper-parameters: C (penalty parameter) and γ (kernel coefficient). Grid search is a brute force method, it trains the model by trying all the possible hyper-parameter combinations within a certain search space previously defined (the grid) and gives as result the model with the combination that gives the best result. It is not a fast nor efficient method, future works will be focused on optimizing the SVM hyper-parameters. Stratified Cross-Validation, with 10 folds, is used to validate the results obtained by the model.

During the Grid Search, the metric to evaluate the models is the Area Under the ROC Curve (AUC) [21]. Which has been chosen due to the unbalance of classes across the datasets. Metrics such as Accuracy are not useful at giving information about the performance of the trained model when the classes are not balanced. Hence, the SVM model which gives the wider AUC for the validation data is chosen as the best for a particular label. A different SVM is used for each label, i.e, long hair and glasses. The Accuracy metric is still used in case of balanced classes.

The control parameters used for the Grid Search are presented at the Table 1. The value 1000 for the parameter C corresponds to the maximum number of possible Cs.

Table 1: Control parameters used to train the SVM models.

C	1, 10, 100, 1000
γ	10^3, 10^2, 10^1, 1, 10^{-1}, 10^{-2}, 10^{-3}

3.4 Evaluation

The results obtained using the DCA+SVM combination are compared to those obtained by HOG+SVM. Stratified Cross-Validation with 10 folds is used as validation method, a variation of the traditional K-Fold Cross-Validation, which preserves the percentage of samples of each class.

4 Datasets

Two datasets are used in this work, one is a custom dataset created by combining samples from five different datasets to train the DCA and the other one is the VIPeR dataset, which is used for classification. Both datasets are composed of bounding boxes containing pedestrians. Thus, the pedestrians detection stage is skipped, for it is a problem that is not within the scope of this work. Details about the configuration of these datasets and the pre-processing are presented in this Section.

4.1 Custom Dataset

Deep Learning methods such as AEs require a large amount of data (thousands or hundreds of thousands samples) in order to obtain satisfactory results. AE learn in an unsupervised manner, which implies that the only issue when creating or obtaining a dataset is the gathering of samples, on the contrary to other supervised Deep Learning methods, which need a large amount of data that also have to be labeled.

For this work, a custom dataset is created to train the DCA, aiming to obtain a considerable amount of images containing pedestrians. The final dataset is a combination of five datasets of pedestrians images, which in total contains 6465 images.

Market1501 [22]: a dataset created for person re-identification composed of several sub-datasets, the one used in this work is the query sub-dataset.

PRID2011 [23]: this dataset contains images of pedestrians from two cameras. Images from both cameras were used. Only images of the camera b are used from the **VIPeR** dataset, since the images from the camera a are used to train and test the classifier.

ETHZ [24]: this dataset provides a large number of images of different pedestrians captured in uncontrolled conditions. However, this dataset provides a large number of images for each person (with invariant soft biometrics). Therefore, the dataset is reduced by taking only 3-4 images of the same person and removing the rest in order to obtain a more heterogeneous set of images.

3DPeS [25]: this dataset contains images of pedestrians mostly within bounding boxes. However, not all images have the same size nor have bounding boxes that fit correctly the pedestrian. To solve this issue, a previously trained model that extracts HOG features from images and applies SVM with a sliding window to extract bounding boxes containing the pedestrian is used. The Python library OpenCV[1] is used for this purpose, which contains the trained HOG+SVM model. Figure 4 shows an original image of the dataset and the bounding box containing the pedestrian extracted using the trained model from OpenCV. This method has some errors: the model sometimes extracts bounding boxes that do crop the pedestrian or that contain too much background. However, these bounding boxes were maintained, so that the network may acquire a higher generalization capability.

[1] Avaliable in `https://github.com/itseez/opencv`

(a) (b)

Fig. 4: Sample images from the 3DPeS and the bounding boxes obtained applying the trained HOG+SVM model from the OpenCV library. Sub-figure *(a)* contains the original images, sub-figure (b) is the bounding box containing the detected pedestrian.

4.2 VIPeR Dataset

This dataset contains images from two cameras, each of which captures one image per person. The samples from the camera a are used to test the performance of our method; the same samples are used to train and test the HOG+SVM model. The dataset contains 632 images per camera and it is labeled with 15 binary attributes [26]. The labels **Long-Hair** and **Sunglasses** are used to test the classifier.

5 Experiments and Results

This section presents the experimental results obtained from two points of view: the reconstruction capability of the DCA and the performance of the SVM according to the feature extractor that was used.

5.1 Autoencoder Reconstruction

Images from the VIPeR (cam a) dataset were used to test the reconstruction capability of the trained DCA. Figure 5 presents two original images from the dataset and their reconstructed version given by the DCA.

(a) (b) (c) (d)

Fig. 5: Original and reconstructed images from the VIPeR dataset. Sub-figures a and c are the original images from the dataset. Sub-figures b and d are the reconstructed images by the DCA.

A visual comparison between the original images and the reconstructed shows that the DCA does not reconstruct with fidelity some aspects of the original images. Information about the colours are not present and the edges are blurred; the reconstructions have less clarity compared to the originals. This happens

because the original images are shrunk so much during the Encoding phase that a lot of information is lost when the dimensionality is reduced. This issue can be solved by saving more information of the original image by setting an architecture that leads to a higher dimensional vector z, so that more information is conserved. However, this work aimed to obtain a relatively low-dimensional feature vector that describes the original image. Therefore, the architecture was kept considering that the reconstruction has a certain level of visual similarity.

5.2 Classification

This section presents the best SVM models for each method (DCA+SVM and HOG+SVM) and their results for some labels of the VIPeR dataset. For the balanced classes, the metric used to optimize the hyper-parameters of the classifier using Grid Search was the Accuracy. For the imbalanced classes, the metric was the AUC. The results achieved by the models correspond to the average accuracy and the standard deviation obtained for the 10 folds of the Stratified Cross-Validation process. A brief exploratory analysis (centered on the class balance) of the dataset is presented for each label before presenting the results.

Experiment #1 Using the label **Long-Hair**, the dataset contains 324 images that belong the class 0 and 308 that belong to the class 1. Since using this label the dataset is almost balanced, the Accuracy was the metric to optimize during the Grid Search in order to obtain the best SVM model. Table 2 presents the results for each model.

Table 2: Results for the label Long-Hair.

	Accuracy (%)
	avg. ± std. dev.
DCA	52.704 ± 4.688
HOG	51.744 ± 5.586

Results show that almost both methods achieved similar performance, with a slight improvement of the average accuracy when using the features extracted by the DCA. The HOG+SVM achieved the worst average accuracy including the highest standard deviation. The low average accuracy achieved by both methods shows that classifying this dataset according to the label Long-Hair is a very hard task.

Experiment #2 The second label that was studied defines if the individuals wear **Sunglasses**. With this label, the dataset contains 517 (81.8%) samples of the class 0 and 115 (18.2%) samples of class 1, which makes it an imbalanced dataset. Thus, the AUC was used as metric for the Grid Search of the SVM. Classifying instances for this label is a very hard task, since using this label the dataset is highly imbalanced and that recognizing if an individual wears

something as small as sunglasses is inherently difficult, even for human beings. Besides, evaluating a classifier for an imbalanced dataset can be tricky. Models usually tend to classify all samples as part of the majority class. Thus, we consider the AUC as evaluation metric. The results are presented in Table 3.

Table 3: Results for the label Sunglasses

	AUC (%)
	avg. ± std. dev.
DCA	56.38 ± 8.72
HOG	52.26 ± 8.66

Results show that DCA+SVM achieved the best average AUC. The results obtained for both labels discussed in the previous sections may lead to consider that using DCA as feature extractor allows to achieve better results than HOG. However, this may not occur for all cases, since DCAs learn features (in an unsupervised manner) that allow to reconstruct the original image but the information that those features contain may not be meaningful for classifying certain labels.

6 Conclusion

The objective of this work was to train a Denoising Convolutional Autoencoder (DCA) using a custom dataset and use the trained network as feature a extractor in combination with a Support Vector Machine (SVM) to solve an image classification task: soft biometrics of pedestrians. The quality of the results obtained with this approach was measured through a comparison with the results obtained by other well-known method for the same dataset.

The proposed goal was accomplished and results show that, for the studied labels, the DCA is an acceptable alternative: the performance of the classifier slightly improves when using the DCA as feature extractor. However, this may not occur for all labels of a dataset, as DCA is not a hand-crafted feature extraction method. We would also like to point out that the results obtained cannot be considered satisfactory. Thus, we conclude that using pure DCAs as feature extractors may not be suitable for supervised classification. Hence, this paper must be considered as the initial step of a work in progress regarding the exploration of the capability of Autoencoders to serve as feature extractors for soft biometrics classification problems. Future work may aim at improving the results by applying a fine-tunning phase during the training of the DCA or training one DCA for each class of the problem.

Future works include testing other network architectures, using different methods to optimize the hyper-parameters of the classifier (such as random search and bio-inspired algorithms) and including supervised learning features

in the network, so that it can be able to learn features that not just allow to reconstruct the original input but also to generate latent representations that can be useful at the classification task.

Acknowledgment

N. M. Romero Aquino thanks the Organization of the American States, the Coimbra Group of Brazilian Universities and the Pan American Health Organization; Author M. Gutoski would like to thank CAPES for the scholarship; Author H.S.Lopes would like to thank to CNPq for the research grant number 440977/2015-0. All authors would like to thank both CAPES and CNPq for the scholarships, as well as NVIDIA and Fundação Araucária.

References

[1] Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (5 2015) 436–444
[2] LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. MIT Press, Cambridge, MA, USA (1998) 255–258
[3] Perlin, H.A., Lopes, H.S.: Extracting human attributes using a convolutional neural network approach. Pattern Recognition Letters **68** (2015) 250 – 259
[4] Levi, G., Hassncer, T.: Age and gender classification using convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2015) 34–42
[5] Zhu, J., Liao, S., Yi, D., Lei, Z., Li, S.Z.: Multi-label CNN based pedestrian attribute learning for soft biometrics. In: 2015 International Conference on Biometrics (ICB). (2015) 535–540
[6] Martinho-Corbishley, D., Nixon, M.S., Carter, J.N.: Retrieving relative soft biometrics for semantic identification. In: 2016 23rd International Conference on Pattern Recognition (ICPR). (2016) 3067–3072
[7] Bottou, L. In: Large-Scale Machine Learning with Stochastic Gradient Descent. Physica-Verlag HD, Heidelberg (2010) 177–186
[8] Wang, X., Guo, R., Kambhamettu, C.: Deeply-Learned Feature for Age Estimation. In: 2015 IEEE Winter Conference on Applications of Computer Vision. (2015) 534–541
[9] Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning (2011) 52–59
[10] Gray, D., Tao, H. In: Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. Springer Berlin Heidelberg, Berlin, Heidelberg (2008) 262–275
[11] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Washington, DC, USA (2005) 886–893

[12] Bengio, Y.: Learning deep architectures for AI. Foundation and Trends in Machine Learning **2**(1) (2009) 1–127
[13] Wang, Z., Bovik, A.C.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE signal processing magazine **26**(1) (2009) 98–117
[14] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. (2008) 1096–1103
[15] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning Temporal Regularity in Video Sequences (2016)
[16] Xu, L., Ren, J., Liu, C., Jia, J.: Deep Convolutional Neural Network for Image Deconvolution. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 1790–1798
[17] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. (2014) 1–15
[18] Nair, V., Hinton, G.: Rectified Linear Units Improve Restricted Boltzmann Machines. In Frnkranz, J., Joachims, T., eds.: Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress (2010) 807–814
[19] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al.: TensorFlow: large-scale machine learning on heterogeneous systems. arXiv:1603.04467 (2016) 1–19
[20] Van Der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors: scikit-image: image processing in Python. PeerJ **2** (2014) e453
[21] Hanley, J.A., Mcneil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143** (1982) 29–36
[22] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable Person Re-identification: A Benchmark. In: 2015 IEEE International Conference on Computer Vision (ICCV). (2015) 1116–1124
[23] Hirzer, M., Beleznai, C., Roth, P.M., Bischof, H. In: Person Re-identification by Descriptive and Discriminative Classification. Springer Berlin, Berlin, Heidelberg (2011) 91–102
[24] Schwartz, W.R., Davis, L.S.: Learning discriminative appearance-based models using partial least squares. In: XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI),, IEEE (2009) 322–329
[25] Baltieri, D., Vezzani, R., Cucchiara, R.: 3DPeS: 3D people dataset for surveillance and forensics. In: Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding, New York, NY, USA, ACM (2011) 59–64
[26] Layne, R., Hospedales, T., Gong, S., Mary, Q., Laboratory, V.: Person re-identification by attributes. In: In British Machine Vision Conference. (2012) 1–11