Learning Local Convolutional Features for Face Recognition with 2D-Warping

Harald Hanselmann and Hermann Ney

Human Language Technology and Pattern Recognition Group,
RWTH Aachen University
surname@cs.rwth-aachen.de

Abstract. The field of face recognition has seen a large boost in performance by applying Convolutional Neural Networks (CNN) in various ways. In this paper we want to leverage these advancements for face recognition with 2D-Warping. The latter has been shown to be effective especially with respect to pose-invariant face recognition, but usually relies on hand-crafted dense local feature descriptors. In this work the hand-crafted descriptors are replaced by descriptors learned with a CNN. An evaluation on the CMU-MultiPIE database shows that in this way the classification performance can be increased by a large margin.

Keywords: Face recognition, 2D-Warping, Convolutional Neural Network

1 Introduction

2D-Warping tries to find a warping mapping between two given images. One of the images serves as source image and the other as target image. The warping mapping assigns each pixel in the source image a matching pixel in the target image. This mapping is optimized according to a given warping criterion (energy function). A similarity measure can be defined based on the assumption that images of the same class can be warped easier than images of different classes. The latter can then be used for e.g. nearest neighbor classification [17]. This approach has achieved very good accuracies for face recognition, especially with large pose variances (e.g. [12]).

The warping criterion is usually composed of two parts, local descriptor similarity and a smoothness term that incorporates the 2D-dependencies of the pixels in a local neighborhood. In the past, mostly hand-crafted features such as SIFT [22] have been used as local descriptors. However, this work aims to learn the local features using a Convolutional Neural Network (CNN). Lately, a lot of different methods learning a feature embedding by using a siamese architecture and a contrastive loss [30, 28, 31, 24]. We use a similar approach to learn a regular grid of local feature vectors that can then be used as input to the 2D-Warping based recognition algorithm.

1.1 Related work

In the past, several methods have been proposed that use 2D-Warping for image recognition [17, 35] or face-recognition in particular [23, 3, 1, 9, 26, 4, 12]. These methods mostly differ in how the warping criterion is defined and optimized, but usually use hand-crafted features such as SIFT [22] to describe the images. In this paper we use the algorithm proposed in [12]. The related work with respect to convolutional features for face recognition and 2D-Warping can be divided into the following parts.

First, there is work related to CNNs and Face Recognition. Similar to many areas of computer vision CNNs also had a large impact on face recognition [13, 37, 33]. While there are approaches taking pose-invariance into account specifically (e.g. [38]), most related to our work are the approaches where a face embedding is learned by optimizing a similarity measure between the face images. This can be done by training a siamese architecture with a contrastive loss [5]. In [28] this has been extended to a triplet-loss (this could be considered a siamese architecture with three streams). The training can be improved by including a classification loss that is trained jointly with the contrastive loss [30], or successively [24]. In [31] additional supervisory signals are included at lower levels of the CNN. However, none of these methods evaluate using features extracted from the lower levels of the CNN in combination with 2D-Warping for face recognition.

Second, there is work related to CNNs and 2D-Warping. The work published in [21] is very closely related to our work. Here the effectiveness of convolutional features for 2D-Warping is demonstrated for the task of keypoint matching with 2D-Warping, but the network is trained in a single-stream architecture without contrastive loss. There has also been research to learn convolutional features for stereo-vision [40, 39]. The CNN is applied on each location resulting in a feature vector as output, which can then be used by a 2D-Warping algorithm as local cost. Usually rectified images are assumed and the warping is done only in the horizontal dimension and no joint training of classification and contrastive loss is applied. In [20] a conditional random field is optimized for depth estimation from a single image. By using a CNN small image patches are mapped to a single depth value which is then used to construct the energy function and the parameters are learned jointly.

Finally, there is work related to CNNs for local descriptor learning. These methods are similar to the methods mentioned in the first part but are not specifically for face recognition and focus on learning patch-similarity. One example is [29] where a siamese architecture is used to learn discriminative features for image patches. In [25] the features are learned unsupervised and in [11] a metric network is included that learns a probability that two input patches are similar.

2 2D-Warping

In this section we briefly review 2D-warping or image matching in compliance with [26]. The approach aims to find a matching between local feature descriptors

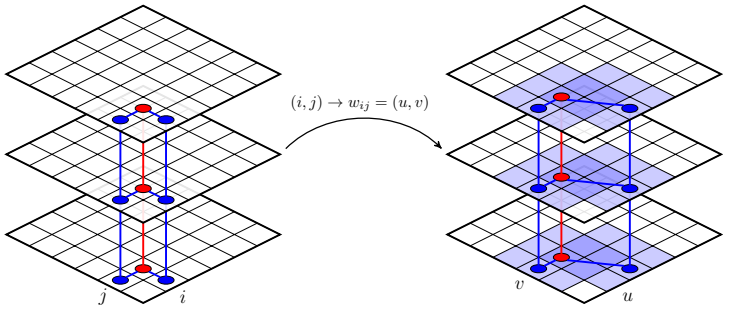


Fig. 1. 2D-Warping: The red feature descriptor is mapped with respect to the horizontal and vertical neighbor. The blue areas indicate the hard monotonicity and continuity constraints.

of two images while respecting a local cost function and neighborhood dependencies. To this end a mapping function is searched that receives a set of local feature vectors of the first image (the source image) as input and maps them to the corresponding set of feature vectors from the second image (the target image). The local features are extracted using a regular grid of dimension $I \times J$ for the source and $U \times V$ for the target image. As a result, the source image is defined as $X \in \mathbb{R}^{I \times J \times D}$ and the target image is defined as $R \in \mathbb{R}^{U \times V \times D}$ where D is the dimension of the feature descriptor. The warping mapping w then maps each pixel of X to a pixel of R:

$$(i,j) \rightarrow w_{i,j} = (u,v) \quad \forall (i,j), \;\; i \in \{1,\ldots,I\}, \;\; j \in \{1,\ldots,J\} \tag{1}$$

A local cost function $dist(\cdot)$ returns a score measuring the similarity of two local descriptors. In this work the l1-norm is used as local cost function. The neighborhood dependencies are realized using a smoothness term $T(\cdot)$ that penalizes large disparities in the mapping of pixels within a local neighborhood. As in [26] we use the l2-norm applied with respect to the vertical and horizontal predecessor as penalty function in addition to hard monotonicity and continuity constraints that limit the possible displacements of neighboring pixels [35]. These choices have been shown to work well for face recognition [26, 12].

Finally, by combining the local cost and the smoothness term the following optimization criterion or energy function can be defined:

$$E(X,R,\{w_{ij}\}) = \sum_{i,j} \left[dist(x_{i,j}, r_{w_{ij}}) + T(w_{i-1,j}, w_{ij}) + T(w_{i,j-1}, w_{ij}) \right]. \tag{2}$$

In the context of classification 2D-Warping can be used in a nearest neighbor classifier to compensate small intra-class variations [17]. Each training (gallery) image is warped to the test (probe) image and the resulting energy is used as a similarity measure. Computing the energy between the probe image and a

large number of gallery images can be costly, but in face recognition often only one frontal-view gallery image per subject is used (mugshot-setup). This case is especially suited for 2D-Warping as a normalized frontal-view image is a very good source image.

Optimizing the energy function in Formula 2 is an NP-complete problem [16] and thus computing the optimal solution is intractable. Therefore, several approximative methods have been proposed (e.g. in the context of face recognition [26, 2, 12]) and the runtime for 2D-Warping depends on the selected algorithm. Here we use the algorithm proposed in [12], since this approach leads to good results for face recognition. The method is called Two-Level Dynamic Programming with lookahead and warprange (2LDP-LA-W). In a local level several candidates for the optimization of a column are computed while on a global level the best sequence of such candidates is found. The procedure is guided by a lookahead that gives a rough estimate of not yet optimized parts of the image and the warprange restricts the possible displacements of each pixel (w.r.t. to their absolute position). The complexity of this algorithm depends on the image and feature dimensions [12]. The local distances are cached in a pre-processing step to avoid multiple computations of the same distance [9]. For the final runtime the choice for the spatial dimensions I,J,U and V is most important. Ideally, they should be kept as small as possible while not sacrificing too much spatial information.

2.1 Features for 2D-Warping

A crucial step in building a nearest neighbor classifier based on 2D-Warping is the choice of the features. In the past mostly handcrafted features such as SIFT [22] have been used. E.g. in [12] the authors extract a SIFT descriptor based on a regular grid. The descriptor is then reduced using PCA [15] and normalized by the l1-norm. For an input with spatial dimension $I \times J$ this results in a 3D structure of dimension $I \times J \times D$ with a D-dimensional feature descriptor at each spatial position $(i, j) \in I \times J$ (c.f. Figure 1). As demonstrated in [21], the output of a single convolutional layer of a CNN can be interpreted in the same way, if the output of a single filter has a dimension of $I \times J$ and the layer has D feature maps. These features can then be used directly to optimize Formula 2. This means the CNN is applied just once on the input image and all local features are extracted directly from the output of one convolutional layer [21].

For face-recognition with the mug-shot setup and a focus on pose-invariance, self-occlusions in the probe images caused by rotations can be compensated by using just the left or the right half of the gallery image [3]. The half-images are generated after the convolutional features have been extracted by simply cutting the feature-maps in half.

3 CNN-Model

Our model is based on a simplified version of the well known GoogLeNet [32]. This deep network implements several 'Inception'-modules, which use parallel

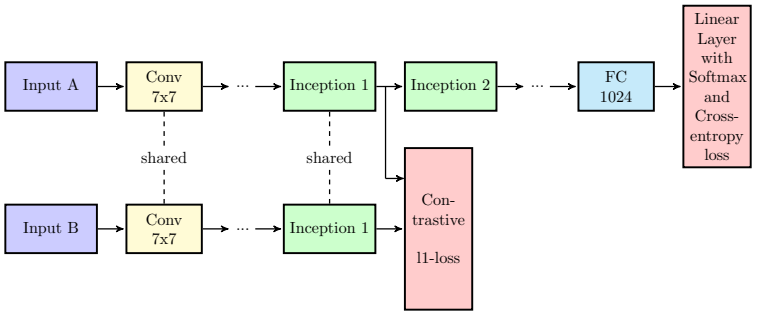


Fig. 2. The siamese CNN model based on GoogLeNet with an early contrastive loss.

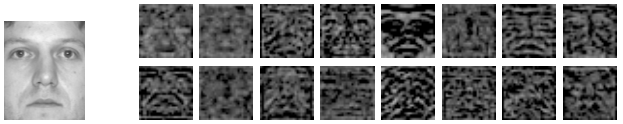
convolutional layers with different kernel-sizes, concatenated to generate one combined output. GoogLeNet uses three classification loss functions at different depths of the network. We simplify the model by using the layers up to the first loss function. This includes the first three inception modules. As 2D-warping needs sufficient spatial dimension to work well, we select the output of one of the earlier convolutional layers as feature input to the 2D-Warping, specifically we use first inception module. The corresponding output is composed of 256 feature-maps with spatial dimension 28×28. As in [30] a contrastive loss is added to the classification loss leading to a siamese architecture [5]. However, we attach the contrastive loss to the output of the first inception module, since this is the layer we will be using later for feature extraction. For such a siamese architecture the training data is composed of positive and negative pairs of images (A, B). For a positive pair the two images have the same class ($A_c = B_c$) while for a negative pair the class differs. We use each image A for the cross-entropy loss while image B serves as additional supervisory signal to minimize positive and maximize negative distances. The final layout of the model is shown in Figure 2.

3.1 Contrastive L1-Loss

The input to the contrastive loss is the output of a convolutional layer with spatial dimension $I \times J$ and D feature maps. The first step is to normalize the features. As described in Section 2, the entries at a specific position in the feature maps are interpreted as the local feature vectors[21]. For this reason we apply a position-wise normalization using the l1-norm:

$$\hat{A}_{i,j,d} = \frac{A_{i,j,d}}{\|A_{i,j}\|_1} = \frac{A_{i,j,d}}{\sum_{d'} |A_{i,j,d'}|} \tag{3}$$

The actual loss function is based on the contrastive loss proposed in [30], but as in [5] we use the l1-distance to calculate the distance between two images and as in [27] a positive margin is included. The local loss for an image pair (A, B) is given by



(a) Input (b) Feature maps

Fig. 3. Learned features using the described architecture. Shown are the features-maps with the highest mean activation (gamma has been increased).

$$L_n(A,B) = \frac{1}{E_{max}} \cdot \begin{cases} \frac{1}{2}\max(0, \|\hat{A}-\hat{B}\|_1 - m_p)^2 & \text{if } A_c = B_c \\ \frac{1}{2}\max(0, m_n - \|\hat{A}-\hat{B}\|_1)^2 & \text{else} \end{cases} \tag{4}$$

where m_p and m_n are margins that regulate the influence of positive and negative pairs. This helps to avoid that the training focuses too much on optimizing pairs of the same class that already have a low distance and pairs of different class that already have a large distance, respectively. This loss function adds two more hyper-parameters to the training procedure, but both can simply be set to the mean distance. Additionally, we normalize the loss using the maximal possible distance E_{max}, which is known at this point due to the position-wise l1-normalization. The contribution of the contrastive l1-loss to the overall loss is weighted by a parameter λ.

Figure 3 shows examples of features learned using the described model. We show the feature maps with the highest mean activation.

4 Experimental evaluation

We implement the contrastive l1-loss with position-wise l1-normalization using the open source framework Caffe [14]. To evaluate the 2D-Warping algorithm 2LDP-LA-W (c.f. Section 2) we use the software provided with [12].

The experimental evaluation is done using the CMU-MultiPIE database [10]. The database contains over 700,000 images recorded in four different sessions and with variations in pose, illumination and facial expression, the former two are especially challenging. There are 15 different poses ranging from $-90°$ to $+90°$ rotation in yaw and 20 different illumination conditions (c.f. Figure 4). There are two special poses emulating a surveillance camera view by including a small amount of tilt. To be able to compare with [12] we use the same pre-alignment, i.e. all images are cropped and normalized based on manual landmarks and pose information using the method proposed in [8].

We use two different settings. Setting 1 is designed to evaluate the method with respect to variations in pose [8]. Images from the first recording session are divided into a training and a test set. The first 100 subjects are used for training (c.f. Section 4.1), while the remaining 149 subjects are used for testing. The illumination is kept constant for this setting which leaves 2086 test images.

(a) (b)

Fig. 4. Example images of the CMU-MultiPIE database. Different poses are shown in (a) and varying illumination is shown in (b).

In setting 2 the method is evaluated with respect to pose and illumination simultaneously. The splitting of train and test subjects is the same as in setting 1, but for setting 2 all illuminations except the front flash (19 in total) are used, while the poses are reduced to those that range from $-45°$ to $+45°$ rotation (6 different poses). Overall there are $16,986$ test images in this setting. In both settings one frontal view image per subject with neutral illumination is used as gallery which means we only have gallery image for each subject (mug-shot setup). In both settings, the trained CNN models have not seen the classes to be recognized before.

4.1 Training

We evaluate several different models, an overview is given in Table 1. The baseline model (CNN-Base) is trained on ImageNet[6] and provided with Caffe [14]. Initializing with these weights we finetune models using the CASIA WebFace[37] and the CMU-MultiPIE database (CNN-ft). From the CASIA WebFace we select the subjects with more than 60 images and create 1000 random pairs (A, B) for each subject. Half of the 1000 pairs are positive, the other half are negative pairs. In total there are $4,874,000$ image pairs for 4874 subjects. Additionally, we use the $30,000$ images from the training set provided by the two settings for the CMU-MultiPIE. For each of the 100 subjects we again generate random pairs and merge this with the training set provided by the CASIA WebFace database. Since the latter has much more classes we use $10,000$ pairs for each subject such that the training set is not dominated too much by the CASIA WebFace database. The model CNN-ft is finetuned using only a cross-entropy loss. This means the model defined in Section 3 is reduced to the first stream and we only use the images A of the image pairs. This model is trained for $1,000,000$ iterations with a batchsize of 32 and a base learning rate of 0.003. The learning rate is reduced gradually with a step-size of $100,000$ iterations. We evaluate the model after each $100,000$ iterations using a small holdout-set from the CMU-MultiPIE images and select the best one for our experiments. Furthermore, a model using both cross-entropy and contrastive loss (CNN-ft-CL) is trained. For this, we use the same number of iterations as the best CNN-ft model and the same hyper-parameters. The additional hyper-parameter λ to weight the contribution of the contrastive loss is set to 0.1.

We also evaluate training the models from scratch using only data from CMU-MultiPIE (CNN-M and CNN-M-CL). Since we have less data than in the

Table 1. Evaluated models.

Name	Training database	Contrastive loss
CNN-Base	Pre-trained on ImageNet[14, 6]	no
CNN-ft	Finetuned from CNN-Base, WebFace, MultiPIE	no
CNN-ft-CL	Finetuned from CNN-Base, WebFace, MultiPIE	yes
CNN-M	MultiPIE	no
CNN-M-CL	MultiPIE	yes

previous case we reduce the number of iterations and the step-size for adjusting the learning rate and evaluation. To have a better starting point for the distances in the contrastive loss, CNN-M-CL is finetuned from the best CNN-M model.

Apart from the different models there are also different ways to extract the features. As mentioned earlier, for 2LDP-LA-W we extract the features after the first inception layer (Inception 1 in Figure 2). For comparison we also extract features at the last layer before the cross-entropy loss is applied, which is a fully connected layer of dimension 1024 (FC 1024 in Figure 2). We use these vectors in a nearest-neighbor classifier with l1-distance (NN-l1) and without 2D-Warping (the spatial dimension is 1×1 at this point). Note that at this point we can not match left and right halves of the gallery anymore, since no spatial information is left.

4.2 Results setting 1: Pose

The first evaluation with respect to robustness against pose variation is done using setting 1. The results are given in Table 2. We specifically compare our approach to the result reported in [12] using 2LDP-LA-W with SIFT features. For the latter, the authors use a 68×86 grid to extract feature vectors of dimension 30 (reduced by PCA). In our case, the feature dimension is with 256 much larger, but the spatial dimension is only 28×28.

It is surprising that applying 2LDP-LA-W with the out-of-the-box features CNN-base already achieves a slightly better result than with the SIFT features, even though no training with respect to a face recognition task has been performed. However, the models trained on face recognition database yield a significant improvement, especially when using the contrastive loss. While training the model from scratch using only the data from the MultiPIE database already achieves a large improvement over the previous approaches, the best result is achieved by finetuning CNN-Base with the WebFace and MultiPIE databases. Using the features extracted at the last fully connected layer does not work as well for setting 1, since the variation in pose can not be compensated, which demonstrates the need for more sophisticated spatial normalization than offered by the pooling layers included in the CNN.

More detailed results for all 14 test poses in setting 1 are given in Figure 5. While most methods achieve close to 100% accuracy on the images with 60°

Table 2. Setting 1: Results reported in accuracy[%].

Method	Total without Surveillance	Total
PLS [8]**	90.5	90.0
MLCE [19]*	92.1	-
2LDP-LA-W + SIFT [12]	90.2	91.5
2LDP-LA-W + CNN-Base	91.7	91.7
2LDP-LA-W + CNN-ft	92.7	93.2
2LDP-LA-W + CNN-ft-CL	**95.5**	**96.1**
2LDP-LA-W + CNN-M	89.3	88.6
2LDP-LA-W + CNN-M-CL	94.0	94.6
NN-l1 + CNN-ft-FC1024	71.6	70.0
NN-l1 + CNN-ft-CL-FC1024	71.9	71.0

* Different pre-alignment.
** Different illumination.

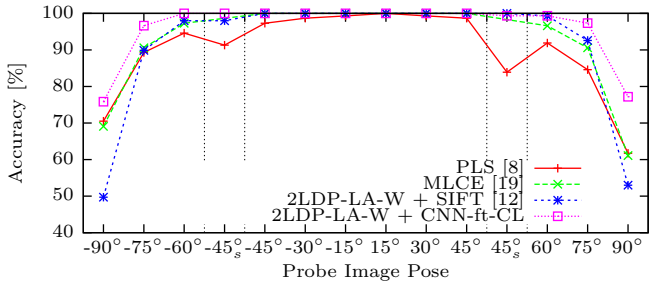


Fig. 5. Setting 1: Results for each pose. The poses marked as -45°_s and 45°_s are the special surveillance poses.

or less rotation in yaw, the more challenging poses prove difficult to handle, especially the poses with 90° rotation. On these two poses our approach with 2LDP-LA-W and CNN-ft-CL features achieves the most improvements.

4.3 Results setting 2: Pose and Illumination

We also evaluate robustness with respect to illumination to see how well the large variations in lighting contained in the CMU-MultiPIE database are learned. The results are given in Table 3. The evaluation only includes the best performing model from setting 1 (also on this task, the other models did not achieve competitive performance). Additionally we evaluated the effect of a normalization with respect to illumination [34], which is also used by the best performing state-of-the-art method [7]. While the features extracted with CNN-ft-CL lead to good

Table 3. Setting 2: Results reported in accuracy [%].

Method	-60°	-45°	-30°	-15°	+15°	+30°	+45°	+60°	Total
Ridge regression [18]*	-	63.5	69.3	79.7	75.6	71.6	54.6	-	-
RL-LDA [41]*	-	67.1	74.6	86.1	83.3	75.3	61.8	-	-
CPF [38]*	-	73.0	81.7	89.4	89.5	80.4	70.3	-	-
AQI-GEM [36]*,**	-	79.0	90.3	97.0	98.3	94.7	87.4	-	-
PBPR [7]*	**90.9**	**97.9**	**99.4**	99.0	**99.9**	**99.2**	**98.2**	**87.8**	**96.6**
2LDP-LA-W									
+ CNN-ft-CL	72.8	79.5	85.8	92.9	95.4	90.1	81.6	71.8	83.7
+ Norm[34] + CNN-M-CL	77.0	91.6	96.1	98.0	99.2	97.6	90.6	79.7	91.2
+ Norm[34] + CNN-ft-CL	85.5	94.7	98.6	**99.5**	99.8	99.0	92.6	86.5	94.5

* Different pre-alignment.
** Automatically detected landmarks.

results, it is evident that the variation with respect to the difficult lighting conditions was not learned to full extend. Normalizing the gallery and test images and finetuning the model further on normalized images (MultiPIE only) yields a significant improvement. Again, training from scratch using only data from the CMU-MultiPIE database performs slightly worse.

5 Conclusion

Using 2D-Warping (2LDP-LA-W in particular) leads to high accuracies for face recognition, especially with respect to pose-invariance. We combined this approach with powerful CNN-models and outperformed 2LDP-LA-W with handcrafted SIFT features by a large margin. This is achieved by using a siamese architecture with a contrastive l1-loss attached to a lower layer of the CNN-model, whose features are the input to the warping algorithm.

For future work it would be interesting to evaluate, if the distance used in the contrastive-layer can be replaced by a warping distance directly. This might lead to problems with the runtime, but a simple warping method such as zero-order warping [17] would be fast enough. It would also be interesting to evaluate if advances such as the triplet loss [28] lead to further improvements.

References

1. Arashloo, S.R., Kittler, J.: Efficient processing of mrfs for unconstrained-pose face recognition. In: IEEE Biometrics: Theory, Applications and Systems (BTAS). pp. 1–8 (2013)
2. Arashloo, S.R., Kittler, J.: Fast pose invariant face recognition using super coupled multiresolution markov random fields on a gpu. Pattern Recognition Letters 48, 49–59 (2014)

3. Arashloo, S., Kittler, J., Christmas, W.: Pose-invariant face recognition by matching on multi-resolution mrfs linked by supercoupling transform. Computer Vision and Image Understanding 115(7), 1073–1083 (2011)
4. Castillo, C.D., Jacobs, D.W.: Wide-baseline stereo for face recognition with large pose variation. In: IEEE CVPR. pp. 537–544 (2011)
5. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE CVPR. vol. 1, pp. 539–546 (2005)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE CVPR. pp. 248–255 (2009)
7. Ding, C., Xu, C., Tao, D.: Multi-task pose-invariant face recognition. IEEE Transactions on Image Processing 24(3), 980–993 (2015)
8. Fischer, M., Ekenel, H.K., Stiefelhagen, R.: Analysis of partial least squares for pose-invariant face recognition. In: IEEE Biometrics: Theory, Applications and Systems (BTAS). pp. 331–338 (2012)
9. Gass, T., Pishchulin, L., Dreuw, P., Ney, H.: Warp that smile on your face: optimal and smooth deformations for face recognition. In: IEEE Automatic Face and Gesture Recognition (FG). pp. 456–463 (2011)
10. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing 28(5), 807–813 (2010)
11. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: unifying feature and metric learning for patch-based matching. In: IEEE CVPR. pp. 3279–3286 (2015)
12. Hanselmann, H., Ney, H.: Speeding up 2d-warping for pose-invariant face recognition. In: IEEE Automatic Face and Gesture Recognition (FG). vol. 1, pp. 1–7 (2015)
13. Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S., Hospedales, T.: When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In: IEEE International Conference on Computer Vision Workshops. pp. 142–150 (2015)
14. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
15. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: IEEE CVPR. vol. 2, pp. II–506 (2004)
16. Keysers, D., Unger, W.: Elastic image matching is NP-complete. Pattern Recognition Letters 24(1-3), 445–453 (2003)
17. Keysers, D., Deselaers, T., Gollan, C., Ney, H.: Deformation models for image recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(8), 1422–1435 (2007)
18. Li, A., Shan, S., Gao, W.: Coupled bias–variance tradeoff for cross-pose face recognition. IEEE Transactions on Image Processing 21(1), 305–315 (2012)
19. Li, S., Liu, X., Chai, X., Zhang, H., Lao, S., Shan, S.: Maximal likelihood correspondence estimation for face recognition across pose. IEEE Transactions on Image Processing 23(10), 4587–4600 (2014)
20. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: IEEE CVPR. pp. 5162–5170 (2015)
21. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: Advances in Neural Information Processing Systems. pp. 1601–1609 (2014)
22. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)

23. Mottl, V., Kopylov, A., Kostin, A., Yermakov, A., Kittler, J.: Elastic transformation of the image pixel grid for similarity based face identification. In: ICPR. pp. 549–552 (2002)
24. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference. vol. 1, p. 6 (2015)
25. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., Schmid, C.: Local convolutional features with unsupervised training for image retrieval. In: IEEE International Conference on Computer Vision. pp. 91–99 (2015)
26. Pishchulin, L., Gass, T., Dreuw, P., Ney, H.: Image warping for face recognition: From local optimality towards global optimization. Pattern Recognition 45(9), 3131–3140 (2012)
27. Sadeghi, F., Zitnick, C.L., Farhadi, A.: Visalogy: Answering visual analogy questions. In: Advances in Neural Information Processing Systems. pp. 1873–1881 (2015)
28. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE CVPR. pp. 815–823 (2015)
29. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: IEEE International Conference on Computer Vision. pp. 118–126 (2015)
30. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems. pp. 1988–1996 (2014)
31. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE CVPR. pp. 1–9 (2015)
33. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: IEEE CVPR. pp. 1701–1708 (2014)
34. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: International Workshop on Analysis and Modeling of Faces and Gestures. pp. 168–182. Springer (2007)
35. Uchida, S., Sakoe, H.: A monotonic and continuous two-dimensional warping based on dynamic programming. In: ICPR. pp. 521–524 (1998)
36. Wu, Z., Deng, W.: Adaptive quotient image with 3d generic elastic models for pose and illumination invariant face recognition. In: Biometric Recognition, pp. 3–10. Springer (2015)
37. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
38. Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., Kim, J.: Rotating your face using multi-task deep neural network. In: IEEE CVPR. pp. 676–684 (2015)
39. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: IEEE CVPR. pp. 4353–4361 (2015)
40. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research 17, 1–32 (2016)
41. Zhu, Z., Luo, P., Wang, X., Tang, X.: Deep learning identity-preserving face space. In: IEEE International Conference on Computer Vision. pp. 113–120 (2013)