Deep Heterogeneous Face Recognition Networks based on Cross-modal Distillation and an Equitable Distance Metric

Christopher Reale[1,2] Hyungtae Lee[1,3] Heesung Kwon[1]
[1]U.S. Army Research Laboratory
[2]University of Maryland, College Park
[3]Booz Allen Hamilton Inc.
reale@umiacs.umd.edu lee_hyungtae@bah.com heesung.kwon.civ@mail.mil

Abstract

In this work we present three methods to improve a deep convolutional neural network approach to near-infrared heterogeneous face recognition. We first present a method to distill extra information from a pre-trained visible face network through the output logits of the network. Next, we put forth an altered contrastive loss function that uses the ℓ_1 norm instead of the ℓ_2 norm as a distance metric. Finally, we propose to improve the initialization network by training it for more iterations. We present the results of experiments of these methods on two widely used near-infrared heterogeneous face recognition datasets and compare them to the state-of-the-art.

1. Introduction

Heterogeneous face recogntion (HFR) is the problem of identifying face images obtained via alternative sensing modalities by matching to a gallery of visible face images. Alternative sensing modalities can be used for a variety of reasons such as poor ambient illumination (infrared), physical limitations (low-resolution), no camera at all (sketch), etc. In this work, we focus on near-infrared (NIR) heterogeneous face recognition.

NIR cameras are widely used in situations where ambient light may not be very bright (e.g. outdoor security cameras at night). To perform face recognition on subjects in these cameras, we would ideally have a gallery of NIR faces to match to. Unfortunately, NIR face galleries generally don't exist, especially with non-cooperative subjects. In contrast, it is usually possible to get a visible light image of just about anyone in the world. The solution to this problem is to match NIR faces to visible galleries with NIR HFR algorithms.

The methods presented in this work build off of the work of Reale et al. [14], which presented a deep convolutional neural network approach NIR HFR. That work takes a two-step approach to training the neural networks. First, an initialization network is trained with a very large dataset to perform visible light face recognition. Then, the HFR networks are initialized with the learned network parameters and further trained on cross-modal data to map visible and infrared faces into a domain independent feature space.

In this work, we improve the performance of the previous approach with three key changes. First, we use cross-modal distillation to leverage more information from the initialization network. In the previous approach, the networks are trained so that images of the same subject are close together and images of different subjects are farther apart. Cross-modal distillation attempts to enforce the following principle: if a subject looks similar/dissimilar to another subject in the visible domain, then the same *degree of similarity* should hold in the IR domain. As shown in Figure 1, this is achieved by training the IR network to reproduce the visible network's logits (classification scores before the softmax output) which can be thought of as similarity scores to the initialization subjects. This allows the IR network to "distill" information from the visible domain network.

The second proposed improvement is to replace the ℓ_2 distance metric with the ℓ_1 distance metric in the contrastive loss during training. This prevents a few feature components from dominating the optimization and allows all the features learned by the initialization network to have comparable effects during the HFR training.

The third improvement we present is to use a better initialization network. We accomplish this by training the initialization network for over twice as many iterations. This yields a more thoroughly trained network with more meaningful, discriminative features for identifying faces.

The remainder of the paper is organized as follows. In Section 2 we discuss other work related to our method. In Section 3 we briefly describe our original method and the changes we have made to improve performance. In Sec-

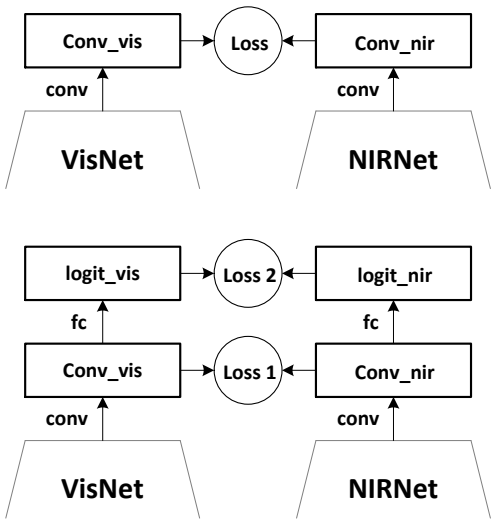

Figure 1. Top: Architecture of Reale et al. [14]. Bottom: Architecture of the proposed network adding a second loss measuring difference of the logits of the two networks to the architecture of [14].

tion 4 we discuss the details of the implementation. In Section 5 we present experiments we performed to evaluate our method. Finally, we conclude the paper in Section 6.

2. Related Work

2.1. NIR HFR

Much research has been published on NIR HFR. Until recently, most approaches used shallow models and/or local features to solve the problem. Klare and Jain [10] use kernel similarities to a set of training subjects as features. Zhu et al. [19] perform domain-adaptive matching with a transductive model and present a new local feature vector. Yi et al. [17] reduce the domain discrepancy locally with restricted Boltzmann machines and then take a subset of PCA coefficients to do the same globally. Jin et al. [5, 6] learn local features that are consistent across sensing modalities yet discriminative within each domain. Juefei-Xu et al. [7] use cross spectral joint dictionaries to reconstruct visible light images from near IR images and vice-versa.

Recently, deep learning has been used to achieve state-of-the-art results for NIR HFR. Virtually all HFR deep learning methods [12, 14, 15] leverage a large visible face dataset to pretrain networks. They then fine-tune them on cross-modal data for HFR.

2.2. Distillation

Distillation was first introduced by Hinton et al. [3] as a means to condense an ensemble of neural networks into a single neural network by extracting information from network logits. It has since been used by Gupta et al. [1] to help train in modalities with limited training data. Su and Maji [16] take a similar approach when training a model for use with low quality data.

3. Our Method

3.1. Original Method

The baseline method [14] works by using deep CNNs to map face images into a domain independent feature space where they can be directly compared across domains. As shown in Figure 2, the networks are trained by minimizing the contrastive loss [2] between visible and IR faces. The contrastive loss is defined as follows,

$$L(\mathbf{x},\mathbf{y}) = \begin{cases} \|\mathbf{x}-\mathbf{y}\|_2^2 & \text{if } l_x = l_y \\ \max(0,(p-\|\mathbf{x}-\mathbf{y}\|_2)^2) & \text{otherwise,} \end{cases}$$

where feature vectors $\mathbf{x}$ and $\mathbf{y}$ have respective labels l_x and l_y, and p is a tuneable parameter. Minimizing this loss encourages feature representations of same-subject faces to be as close together as possible and feature representations of different-subject faces to be farther apart.

In order to speed-up the optimization and reduce the amount of cross-modal training data needed (which is generally not plentiful), the networks are pretrained to perform visible face recognition. In our case, we first train the networks to recognize 10,575 subjects in the CASIA WebFace dataset [18]. We then tweak the networks to perform heterogeneous face recognition.

3.2. Cross-modal Distillation

The first method we use to improve performance is cross-modal distillation. The main assumption is that if a subject looks similar or dissimilar to another subject in visible imagery, that should carry over into the near-infrared imagery. Cross-modal distillation does this by using information about the initialization subjects in the pretrained network. While that information is not included in any dataset, we can infer it from the initialization network output logits.

The initialization network is trained to recognize 10,575 celebrities from their face images. Given a face, this is done by calculating similarity scores (sometimes called logits) for each possible celebrity in the dataset. The network classifies the face image as belonging to the celebrity with the highest score. The network can perform this operation on any face image, regardless of the subject in the image. While it is not useful for recognition of subjects which are

not in the initialization set (like our test and training set subjects), the logits still contain valuable information about the appearance of the face. They can be thought of as a rough estimate for how similar the input face is compared to the celebrity faces in the visible domain. A high logit indicates the input face looks very similar to the celebrity while a low logit indicates the opposite.

As shown in Figure 2, we incorporate this extra information into our algorithm by adding a second contrastive loss function that takes the logits as inputs. This contrastive loss operates only on same-subject pairs, so it essentially performs regression on the logits. Basically, we're trying to ensure that the infrared network can recreate the logits that are generated by the visible network. The idea is that in order to accomplish this task, the network is forced to learn more disciminative features in the convolutional layers of the network. Also note that, as with the convolutional weights, the weights of the fully connected layers are initialized with the corresponding values in the initialization network. Without the pretrained values, the computed logits would be meaningless.

3.3. Contrastive Loss Metric

The second adjustment we make to improve performance is a change in the metric used by the contrastive loss. In the original work, only the euclidean norm is used with the contrastive loss. On the other hand, when testing, Reale et al. [14] cross-validated over three distance metrics to compare face images (ℓ_1, ℓ_2, and cosine). We thought it might be beneficial to cross-validate over the distance metric used in the contrastive loss as well.

In this work, we only cross-validate over the ℓ_1 and ℓ_2 norms. When using the ℓ_1 norm, the constrastive loss is computed as follows,

$$L(\mathbf{x},\mathbf{y}) = \begin{cases} \|\mathbf{x}-\mathbf{y}\|_1 & \text{if } l_x = l_y \\ \max(0, p - \|\mathbf{x}-\mathbf{y}\|_1) & \text{otherwise,} \end{cases}.$$

This gives the training algorithm more flexibility to tailor the networks for the data. It turns out that in virtually all experiments, the ℓ_1 norm provides the best results. We believe it tends to be a better fit for this data because it prevents a few feature components from dominating the optimization. By this we mean that the gradient passed backwards by a given feature component no longer depends on the components magnitude. For example, consider the ℓ_2 norm. One component of gradient of the ℓ_2 norm squared is equal to that component itself. Because of this, components that naturally have a larger discrepancy across domains will have an undue effect on the network parameters when their gradients back-propagated. The gradient of the ℓ_1 norm, on the other hand, is merely the sign of each component and does not depend on the magnitude of the components at all. This prevents a few components from dominating the optimization, allowing for a better solution.

Since we are using two contrastive loss functions (the original one and the one for distillation), we also used cross-validation to determine the distance metric for the distillation loss function. In that case, the ℓ_2 distance yielded better results.

3.4. Improved Initialization

The third adjustment we make to improve performance is by providing the networks with a better initialization. Generally, a simple way to improve performance of any deep network is to train it for longer. Usually this will achieve a small increase in performance. In our case, extra training of the HFR networks does not change the performance much because of the limited amount of training data available.

While we do not have enough cross-modal training data to warrant training the HFR networks for more iterations, we do have a large initialization dataset (CASIA Webface). Thus we train the initialization network for a longer period of time (450,000 iterations vs 200,000 iterations). This makes the initialization learn more effective features for recognizing faces (83% vs 80% recognition rate on a validation set from CASIA Webface) and therefore increases the HFR performance.

4. Engineering Details

4.1. Network Structure

We use the same network structure from [14]. Shown in Table 1, it consists of five sections, each of which contains two convolution layers and a max-pooling layer (with the exception of the last section which uses average pooling). All convolution layers feature 3×3 filters and are followed by rectified linear units (ReLU). After the five sections, there is one fully connected layer that serves as a softmax classifier. In our previous work, we discarded the softmax layer after initialization, but now it is needed for cross-modal distillation as described in Section 3.2.

4.2. Image Preprocessing

We follow [14] and perform very minimal image preprocessing. We first align the face images with the Dlib [9] implementation of a regression tree face alignment algorithm [8]. This method works for both NIR and visible images despite only being trained on the latter. We crop and resize the faces to be 100×100 pixel square images. Finally, we convert the images to gray-scale and subtract the mean face image (calculated from the WebFace dataset). Figures 3 and 4 show sample images for the CASIA WebFace and NIR-VIS 2.0 datasets respectively.

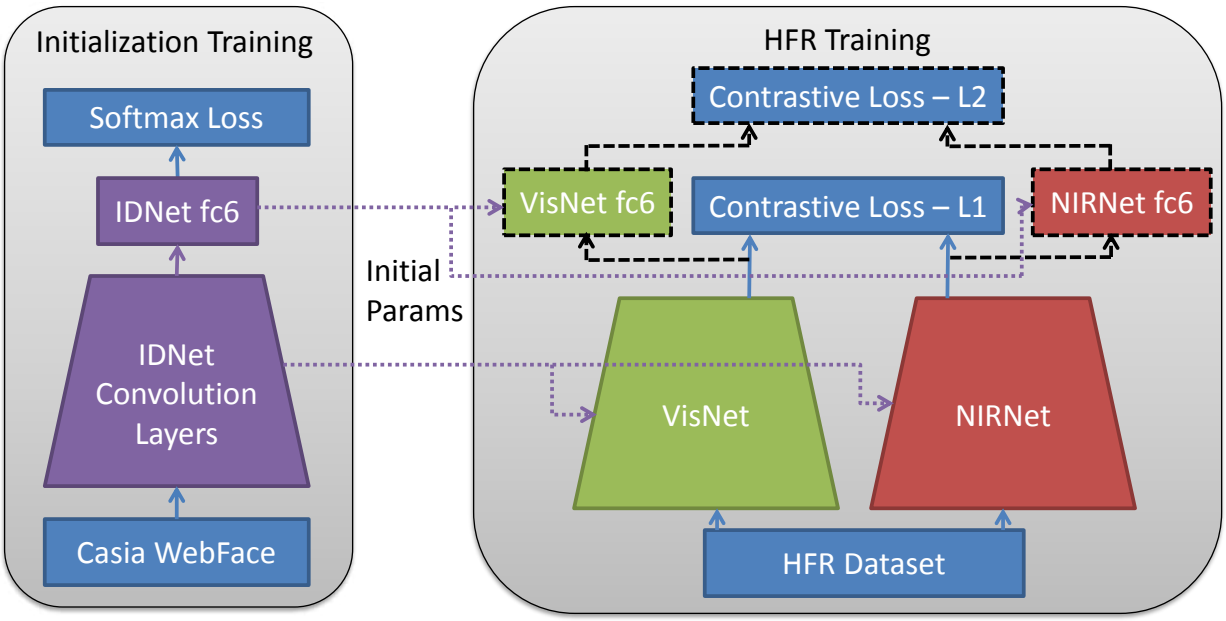

Figure 2. Network Diagram: This figure shows a visualization of our training algorithm and how our improvements affect it. The gray section on the left shows the initialization training setup. One of the contributions of this work is to improve this initialization by training it for longer. The gray section on the left shows the HFR training. Black dashed lines indicate any components added to include cross-modal distillation.

Figure 3. Sample Webface images.

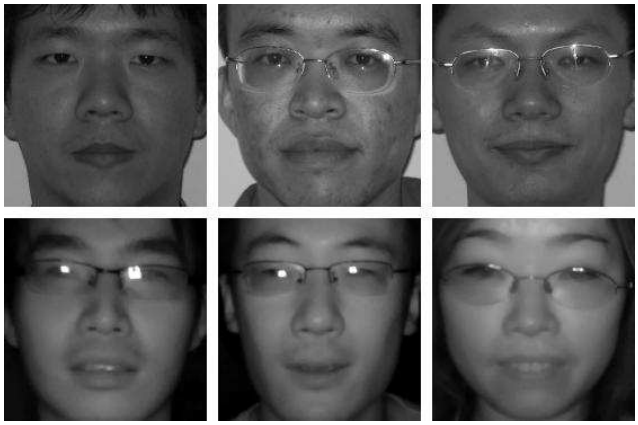

Figure 4. Sample NIR-VIS 2.0 face images. The top row is visible-light and the bottom row is near-infrared.

4.3. Training Details

4.3.1 Initialization Training

We train the initial network using the Caffe [4] deep learning framework for 450,000 iterations (the baseline method trains for 200,000 iterations) with a batch size of 256 images. We initially set the learning rate to be .01 and reduce it by a factor of 10 after 350,000 and 400,000 iterations. We set the momentum to .9 and the weight decay to .0005. We use a single NVIDIA 12GB GeForce GTX Titan X GPU to train IDNet which takes approximately four days.

Table 1. Face recognition network layer details

	Name	Type	Filter Size	Stride	Output Size	Params
Section 1	conv11	Convolution	$3 \times 3 \times 32$	1	$100 \times 100 \times 32$	288
	relu11	ReLU			$100 \times 100 \times 32$	0
	conv12	Convolution	$3 \times 3 \times 64$	1	$100 \times 100 \times 64$	18.4K
	relu12	ReLU			$100 \times 100 \times 64$	0
	pool1	Max Pooling	2×2	2	$50 \times 50 \times 64$	0
Section 2	conv21	Convolution	$3 \times 3 \times 64$	1	$50 \times 50 \times 64$	36.7K
	relu21	ReLU			$50 \times 50 \times 64$	0
	conv22	Convolution	$3 \times 3 \times 128$	1	$50 \times 50 \times 128$	73.7K
	relu22	ReLU			$50 \times 50 \times 128$	0
	pool2	Max Pooling	2×2	2	$25 \times 25 \times 128$	0
Section 3	conv31	Convolution	$3 \times 3 \times 96$	1	$25 \times 25 \times 128$	111K
	relu31	ReLU			$25 \times 25 \times 128$	0
	conv32	Convolution	$3 \times 3 \times 192$	1	$25 \times 25 \times 192$	166K
	relu32	ReLU			$25 \times 25 \times 192$	0
	pool3	Max Pooling	2×2	2	$13 \times 13 \times 192$	0
Section 4	conv41	Convolution	$3 \times 3 \times 128$	1	$13 \times 13 \times 128$	221K
	relu41	ReLU			$13 \times 13 \times 128$	0
	conv42	Convolution	$3 \times 3 \times 256$	1	$13 \times 13 \times 256$	295K
	relu42	ReLU			$13 \times 13 \times 256$	0
	pool4	Max Pooling	2×2	2	$7 \times 7 \times 256$	0
Section 5	conv51	Convolution	$3 \times 3 \times 160$	1	$7 \times 7 \times 160$	369K
	relu51	ReLU			$7 \times 7 \times 160$	0
	conv52	Convolution	$3 \times 3 \times 320$	1	$7 \times 7 \times 320$	461K
	relu52	ReLU			$7 \times 7 \times 320$	0
	pool5	Avg Pooling	7×7	1	$1 \times 1 \times 320$	0
	fc6	Fully Connected			10575	3.38M
	cost	Softmax			10575	0

4.3.2 HFR Training

We train the HFR networks with the same setup as in [14] with a few notable differences due to the incorporation of our new methods. First, due to the large number of parameters in the fully connected layers (see Table 1), we do not allow them to be adjusted during the HFR optimization. Additionally, we found that when training with distillation on the HFB dataset, we did not have to fix as many of the convolutional layers (i.e. the model didn't overfit as easily with distillation). We attribute this to relatively low number of training subjects (100) in the HFB dataset. The additional information from the CASIA WebFace subjects helped to improve generalization.

5. Experiments and Results

5.1. Datasets

We test our algorithms on two widely-used NIR HFR datasets: CASIA HFB and CASIA NIR-VIS 2.0. Both datasets are organized in the same manner by splitting into two views: View1 for parameter selection and View2 for evaluation. View1 consists of a single experimental setup, whereas View2 contains 10 different setups, the results of which are averaged. In CASIA HFB, the typical experiment setup splits the set of subjects into training and testing groups. This yields about 1000 visible and 1500 NIR images for training and similarly 1000 visible and 1500 NIR images for testing. In NIR-VIS 2.0, 715 subjects are split into training and testing groups, with about 2500 visible and 6000 NIR images for training. CASIA NIR-VIS 2.0 is slightly different in that it restricts algorithms to one gallery image per subject during testing. So while there are about 6000 NIR images for testing, there are only 358 gallery images to compare to.

In addition to different numbers of images, some of the images in NIR-VIS 2.0 present challenging variations such as difficult poses, whereas the HFB images were mostly captured in a more controlled environment. Combined with the higher number of subjects and restriction of one image per gallery subject, this makes the NIR-VIS 2.0 dataset significantly more challenging.

HFB	Rank 1	FAR=.01	FAR=.001
Baseline [14]	97.58	96.9	85.0
Distill	98.55	94.7	81.2
L1 Loss	98.68	95.0	82.6
Better Init.	99.08	97.7	87.7
All Three	**99.52**	98.6	91.8

Table 2. Performance of our algorithms on View2 of CASIA HFB Face Database

NIR-VIS 2.0	Rank 1	Std. Dev.	FAR=.001
Baseline [14]	87.1	0.88	74.5
Distillation	87.5	1.04	76.1
L1 Loss	89.4	1.23	79.5
Better Init.	90.8	0.79	77.9
All Three	**92.6**	0.64	81.6

Table 3. Performance of our algorithms on View2 of CASIA NIR-VIS 2.0 Face Database

HFB	Rank 1	FAR=.01	FAR=.001
IDNet	80.9	70.4	36.2
P-RS [10]	87.8	98.2	95.8
C-DFD[11]	92.2	85.6	65.5
THFM [19]	99.28	99.66	98.42
[17]	99.38	-	92.25
Our Method	**99.52**	98.6	91.8

Table 4. Performance comparison to other algorithms on View2 of CASIA HFB Face Database

NIR-VIS 2.0	Rank 1	Std. Dev.	FAR=.001
C-CBFD[13]	81.8	2.3	47.3
[15]	85.9	0.9	78.0
[17]	86.2	0.98	81.3
[12]	**95.74**	0.52	91.03
Our Method	92.6	0.64	81.6

Table 5. Performance comparison to other algorithms on View2 of CASIA NIR-VIS 2.0 Face Database

5.2. Results

We present the results of evaluations of five variants of our method (each method individually, all three together, and the original method as a baseline) in Tables 2 and 3. From the results, it is clear that all three methods have a positive effect on the recognition performance.

Of the three methods, distillation provides a more modest increase in performance, though it is comparable to the other two on the HFB dataset. We attribute this to the smaller number training subjects and images in HFB. This makes information about additional subjects (through distillation) more valuable. On the other hand, NIR-VIS 2.0 provides more training subjects and images. Thus, the training algorithm does not benefit as much from information about additional subjects. The other two methods (ℓ_1 metric and better initialization) help the training on both datasets, with the better initialization adding a noticeably bigger performance boost in both cases. This demonstrates the remarkable ability of deep networks to soak up information even after they have already been trained for a long time.

Finally, it is clear that the best performance is achieved when all three methods are used together. The recognition rates are improved from 97.58% to 99.52% and from 87.1% to 92.6% on the HFB and NIR-VIS 2.0 datasets respectively. Additionally, the verification rates increase from 85.0 to 91.8 and from 74.5 to 81.6.

We compare the results of our method to those of other methods in Tables 4 and 5. Our method performs very well on the HFB dataset with the highest recognition rate among all algorithms. This should be taken with a grain of salt as some papers don't report results on this dataset. On the other hand, our method does not achieve state-of-the-art results on NIR-VIS 2.0. Although it is not better than all other published methods, it is second best and within a few percentage points.

6. Conclusion

In this work we have presented three improvements for NIR HFR. We have evaluated each individually and shown that they all perform better than the original baseline method. Additionally, we have shown that the combination of all three methods performs even better. We have also compared our combined method with the current state-of-the-art.

References

[1] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836, June 2016. 2

[2] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006. 2

[3] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[5] Y. Jin, J. Lu, and Q. Ruan. Coupled discriminative feature learning for heterogeneous face recognition. *Information Forensics and Security, IEEE Transactions on*, 10(3):640–652, 2015. 2

[6] Y. Jin, J. Lu, and Q. Ruan. Large margin coupled feature learning for cross-modal face recognition. In *Biometrics (ICB), 2015 International Conference on*, pages 286–292, May 2015. 2

[7] F. Juefei-Xu, D. Pal, and M. Savvides. Nir-vis heterogeneous face recognition via cross-spectral joint dictionary learning and reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 141–150, 2015. 2

[8] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1867–1874. IEEE, 2014. 3

[9] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 3

[10] B. F. Klare and A. K. Jain. Heterogeneous face recognition using kernel prototype similarities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1410–1422, 2013. 2, 6

[11] Z. Lei, M. Pietikainen, and S. Li. Learning discriminant face descriptor. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):289–302, Feb 2014. 6

[12] X. Liu, L. Song, X. Wu, and T. Tan. Transferring deep representation for nir-vis heterogeneous face recognition. In *2016 International Conference on Biometrics (ICB)*, pages 1–8, June 2016. 2, 6

[13] J. Lu, V. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(10):2041–2056, Oct 2015. 6

[14] C. Reale, N. M. Nasrabadi, H. Kwon, and R. Chellappa. Seeing the forest from the trees: A holistic approach to near-infrared heterogeneous face recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 320–328, June 2016. 1, 2, 3, 5, 6

[15] S. Saxena and J. Verbeek. Heterogeneous face recognition with cnns. In *Computer Vision–ECCV 2016 Workshops*, pages 483–491. Springer, 2016. 2, 6

[16] J.-C. Su and S. Maji. Cross quality distillation. *arXiv preprint arXiv:1604.00433*, 2016. 2

[17] D. Yi, Z. Lei, and S. Li. Shared representation learning for heterogenous face recognition. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–7, May 2015. 2, 6

[18] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 2

[19] J.-Y. Zhu, W.-S. Zheng, J.-H. Lai, and S. Li. Matching nir face to vis face using transduction. *Information Forensics and Security, IEEE Transactions on*, 9(3):501–514, March 2014. 2, 6