Seeing the Forest from the Trees: A Holistic Approach to Near-infrared Heterogeneous Face Recognition

Christopher Reale[1,2] Nasser M. Nasrabadi[3] Heesung Kwon[1] Rama Chellappa[2]
[1]U.S. Army Research Laboratory
[2]University of Maryland, College Park
[3]West Virginia University
reale@umiacs.umd.edu heesung.kwon.civ@mail.mil
nasser.nasrabadi@mail.wvu.edu rama@umiacs.umd.edu

Abstract

Heterogeneous face recognition is the problem of identifying a person from a face image acquired with a non-traditional sensor by matching it to a visible gallery. Most approaches to this problem involve modeling the relationship between corresponding images from the visible and sensing domains. This is typically done at the patch level and/or with shallow models with the aim to prevent overfitting. In this work, rather than modeling local patches or using a simple model, we propose to use a complex, deep model to learn the relationship between the entirety of cross-modal face images. We describe a deep convolutional neural network based method that leverages a large visible image face dataset to prevent overfitting. We present experimental results on two benchmark datasets showing its effectiveness.

1. Introduction

In recent years, a significant amount of research in the computer vision community has focused on Heterogeneous Face Recognition (HFR) [21]. The objective of HFR is to be able to perform face recognition with probe images captured via alternative sensing modalities. Due to the ubiquity of visible cameras, virtually all face galleries are comprised of visible light images. Thus the main challenge of HFR is enabling the cross-domain comparison of probe images to visible-light gallery images. Most works address this problem by selecting [14, 35] or learning [11, 32] features that, among other things, are more invariant across domains than raw pixels. While these methods achieve some degree of success, due to data constraints, the features used are almost always local in nature or learned using shallow models.

Considering that most HFR datasets have large feature dimension (at least 100x100 pixels) and only a moderate number of images for training (fewer than 10000), any attempt to learn global features with deep models will likely overfit to the training set. To get around this problem, HFR algorithms learn features to represent patches rather than whole images. This alleviates the data constraints in two ways: by increasing the number of training samples (because there are multiple patches per image) and by decreasing the feature dimension (because patches are smaller than whole images). Although some works do consider global features [11, 32], the limited training data restricts them to using simple models.

Although learning local features and using simple models make HFR more tractable, recent computer vision research has shown that global features and deep models tend to outperform local features and shallow models. Even just using a simple spatial pyramid approach (i.e. a naive globalization of a local histogram feature) can improve system performance for many applications [16]. This is especially true for current state-of-the-art algorithms in face recognition, which use deep networks to extract global representations for face images. This allows for a richer representation that can model the face as a whole rather than as a collection of parts. In this work, we show how to use deep learning to leverage a large visible face recognition dataset to learn global features for HFR.

Deep networks perform well due to their ability to learn information from extremely large (sometimes unlabeled) datasets. In fact, the recent surge in deep learning research was spurred in part by the breakthrough method of Hinton et al. [5] to initialize deep networks. They proposed to greedily pretrain the layers of a network with a generative model. This helps the stochastic gradient descent learning algorithm by guiding it towards better local minima [3]. In the case where there is limited training data, it also helps to prevent overfitting. In this work, we propose to adapt the same paradigm to HFR. Like [5], we initialize

networks using the abundant data (labeled visible faces) and then tweak them with the scarcer application-specific data (labeled/corresponding visible and infrared faces).

More specifically, our approach is as follows. We first train a deep convolutional neural network on a large visible face dataset for identification. We then use the trained network to initialize networks that will be used to extract features from visible and near-infrared images for HFR. Finally, we further optimize the HFR networks on the HFR training data to couple their output features, making them suitable for cross-modal face recognition.

The main contributions of this paper are as follows:

- We present the first method to use a deep model to learn global features for HFR.
- We learn coupled deep convolutional neural networks to map visible and NIR faces into a domain-independent latent feature space where they can be compared directly.
- We show how to leverage a large visible face recognition dataset to prevent overfitting of the networks.

The remainder of the paper is organized as follows. In Section 2 we review relevant HFR, face recognition, and deep learning literature. In Section 3 we describe our approach. In Section 4 we provide implementation and engineering details. In Section 5 we present experimental results and discuss their implications. In Section 6 we conclude the paper.

2. Related Work

2.1. NIR-Vis Face Recognition

HFR has been extensively researched over the past decade, with NIR being one of the most prominent alternative sensing modalities. Here we briefly describe some recent works on the subject. Klare and Jain [15] use kernel similarities to a set of training subjects as features. Zhu et al. [35] propose a new feature descriptor and a transductive model for domain-adaptive matching. Yi et al. [32] use restricted Boltzmann machines to reduce the domain difference locally and ignore initial PCA coefficients to do the same globally. Jin et al. [9, 10] learn local features to represent images consistenly across domains and discriminatively within each domain. Juefei-Xu et al. [11] use cross spectral joint dictionaries to reconstruct visible light images from near IR images and vice-versa. They then compare images directly. A common drawback of all these methods is they do not use a deep, global representation of face images, which has been shown to produce superior results for face recognition.

2.2. Deep Learning Face Recognition

Face recognition has been researched heavily for decades [34]. With few exceptions ([24]) recent state-of-the-art algorithms have been dominated by deep convolutional neural networks trained on extremely large datasets to produce global feature representations. Taigman et al. [30] were the first to train a deep neural network for face recognition. They trained on the private Social Face Classification dataset which contains 4.4 million labeled images of 4030 subjects. They used convolutional, locally connected, and fully connected layers in their network. Additionally they fine-tune the parameters for verification with a siamese network. They extended their work in [31] and increased the dataset size to 500 million images of 10 million subjects. Sun et al. [25, 26, 27, 28] make a variety of adjustments to improve the performance of deep networks for face recognition including a joint verification-identification loss function, different network architectures, and Bayesian metric learning. They use the private CelebFaces [25] (202,599 images of 10,177 subjects) and WDRef [1] (99,773 images of 2,995 subjects) datasets to train their networks. They also use convolutional, locally connected, and fully connected layers. Parkhi et al. [22] learn a feature embedding by using a triplet loss function. They also detail the steps they took to create their 2.6 million image dataset. Schroff et al. [23] leveraged a 200 million images of 8 million subjects to train a network. This has the best performance to date on Labeled Faces in the Wild (LFW) [6], a standard unconstrained face recognition benchmark.

3. Our Method

3.1. Network Structure and Initialization

The network structure we use throughout this work (detailed in Table 1) is from the Googlenet [29] family of networks (i.e. deep with small convolutional filters). It is comprised of five major sections connected in series. Each section contains the following in order: a convolutional layer, a rectified linear unit layer, a second convolutional layer, a second rectified linear unit layer, and a max pooling layer. The only exception is that the fifth (and last) pooling layer is an average pooling layer. The output of the last section is fed to a fully connected layer whose output is evaluated by a softmax loss function for identification. We composed the network entirely of convolutional layers (as opposed to locally and fully connected layers) to minimize the number of parameters so as to reduce the risk of overfitting. This is not a concern during the initial training, but can become an issue when tweaking the network for HFR due to limited HFR training data.

We train the network with the standard stochastic gradient descent (SGD) and backpropagation on the CASIA WebFace Database [33]. It is comprised of 494,414 images

of 10,575 subjects, which is large enough to provide sufficient generalization. We held out 10,000 images for validation purposes and, after training, the network was able to classify 80 percent of them correctly. We also tested the network on Labeled Faces in the Wild (using the negative distances of pool5 layer features as similarity scores) and achieved 96 percent accuracy. While not state-of-the-art on LFW, the network is more than good enough to serve as an initialization for our HFR application. Within-domain identification experiments on the HFR training data yield greater than 99.5 percent accuracy for both visible and near-infrared images. Furthermore, it has significantly fewer parameters than state-of-the-art networks, which lowers the risk of overfitting when adapting the network for HFR. For the remainder of this paper, we will refer to this network as IDNet.

3.2. HFR Networks

While we would ultimately like to learn a network for identification on infrared faces, we do not have access to infrared face images of probe subjects for training. Instead, we train two networks for cross-modal verification: VisNet (for visible images) and NIRNet (for near infrared images). We initialize these networks as copies of IDNet, with the exclusion of the fully connected softmax classifier. The fully connected layer contains more parameters than the rest of the network combined, so removing it helps to prevent overfitting. Additionally, the outputs of the fully connected layer are each highly tuned for a specific subject in the WebFace dataset, and are not likely to generalize well to arbitrary subjects. After removing layer fc6, the last layer of each network is pool5 and the output of each is one 320-channel pixel.

We train VisNet and NIRNet to couple their output features by creating a siamese network [2, 4] as shown in Figure 3. Although we initialize them with the same values, unlike the normal use of a siamese network, we do not force the two halves of the network to share weights during training. This allows the networks more freedom to capture features that manifest differently in visible and NIR images. We use the contrastive loss [4] of the outputs of the VisNet and NIRnet as the loss function for the network. The contrastive loss L on vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ can be written as follows,

$$L(\mathbf{x}, \mathbf{y}) = \begin{cases} \|\mathbf{x} - \mathbf{y}\|_2^2 & \text{if } l_x = l_y \\ \max(0, (p - \|\mathbf{x} - \mathbf{y}\|_2))^2 & \text{otherwise} \end{cases} \quad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ have respective labels l_x and l_y, and p is a tuneable parameter. Minimizing the contrastive loss makes the distance between images of the same subject as small as possible while ensuring images of different subjects are at least a distance of p away from each other.

Figure 1: Aligned and cropped Webface images.

As with IDNet, we train using stochastic gradient descent and calculate the gradients using the backpropagation algorithm. Whereas in the IDNet training a sample consists of a visible image and its corresponding identity label, here, a sample consists of a visible image, an infrared image, and a binary label indicating whether the images depict the same subject.

4. Implementation Details

4.1. Image Preprocessing

We perform very minimal image preprocessing. We first align all faces using [12] from the Dlib C++ library [13]. This method works well on NIR images in addition to visible ones. Next we crop the faces and resize them to be 100×100 pixels. We then convert the images to gray-scale and subtract the mean face image of the WebFace dataset. We do not scale or filter the images in any way. Sample aligned and cropped images (prior to mean subtraction) are shown in Figures 1 and 2.

4.2. IDNet Details

We train IDNet using the Caffe [8] deep learning framework 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 every 80,000 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 two days.

4.3. HFR Details

When training the HFR networks, we are given training data consisting of a set of labeled visible faces and a set of labeled infrared faces. None of the subjects in the testing set are present in the training set. We use all possible same-subject visible-infrared image pairs as positive samples. Be-

	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

Table 1: IDNet layer details

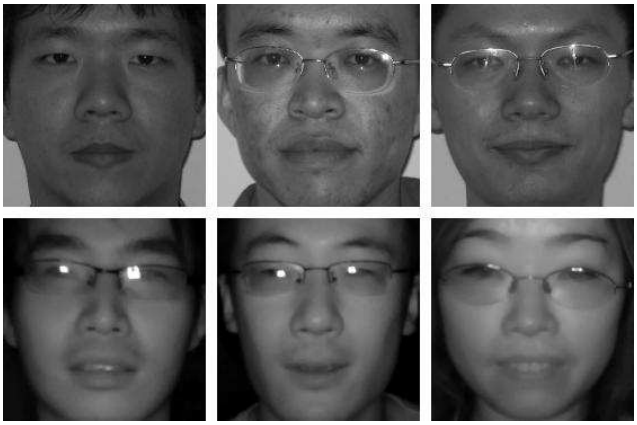

Figure 2: Aligned and cropped NIR-VIS 2.0 face images. The top row is visible-light and the bottom row is near-infrared.

cause there are significantly more different-subject image pairs, we do not use all of them, but rather we balance our training set by including the same number of negative samples as positive samples. We choose the different-subject pairs randomly.

While training IDNet from scratch takes two days, tweaking VisNet and NIRNet can be done much quicker with IDNet as a good initialization point. This allows us to choose some parameters by cross-validation. For all experiments we use a batch size of 64 image pairs. For the learning rate policy, we halve the learning rate every 1000 iterations and set the momentum to .9. We found that both the loss function and performance saturated before 4,000 iterations, allowing us to train the network in about 20 minutes. Parameters we choose with cross-validation include the initial learning rate, the weight decay, the contrastive loss parameter, and the distance metric used for nearest neighbor classification (ℓ^2-norm, ℓ^1-norm, or cosine).

Although we generate as many training samples as possible while maintaining a balanced dataset, the HFR network training is still prone to overfitting due to too little data and too many parameters. We alleviate the overfitting problem by forcing the convolution kernels of some of the network

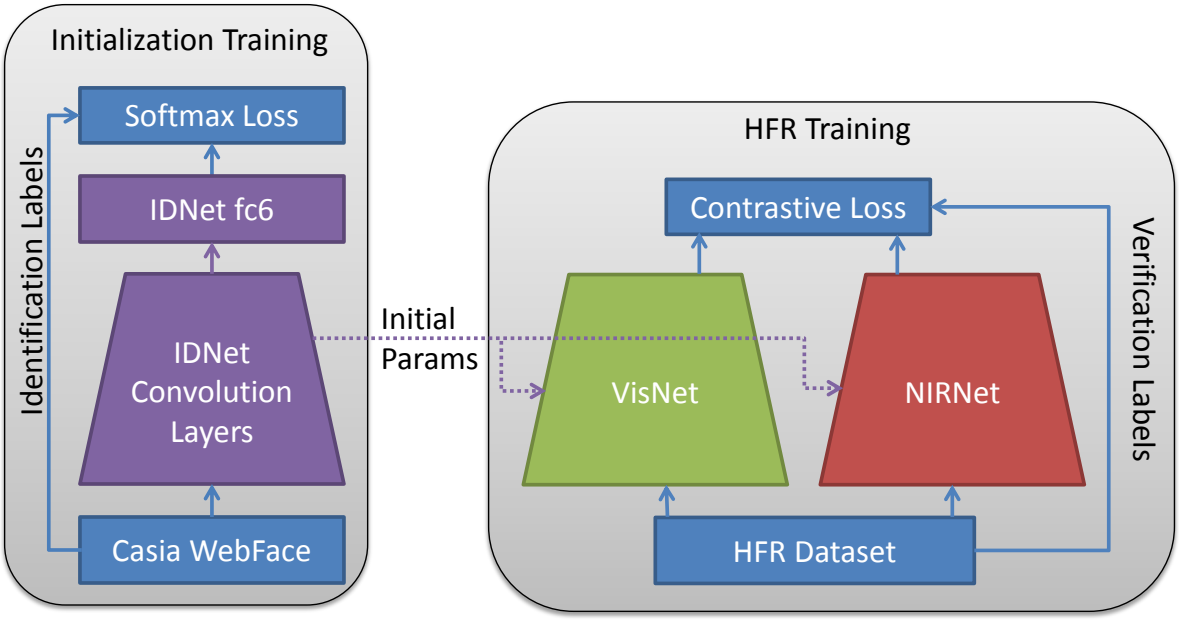

Figure 3: Network Diagram

layers to maintain the original IDNet values. This reduces the number of parameters the training algorithm must learn, lessening the amount of training data needed.

Ideally we would determine which layers to fix entirely through cross-validation. Unfortunately, with each half of the network having 10 convolutional layers, there are $2^{10} \times 2^{10} \approx 10^6$ combinations to test (where a combination indicates for each layer whether it will be fixed to its initial value or allowed to float during training). Instead, we use intuition to select a small subset and cross-validate over those. We first assume that if one layer in a section is fixed, then the other must be fixed as well (i.e. if conv32 in VisNet is fixed, then conv31 in VisNet must also be fixed). We then further assume that at least one of the first and last sections must be floating (i.e. not fixed) and that all floating layers in a network must form a contiguous block. Our assumptions reduce the combinations from more than a million down to the 81 shown in Table 4.

5. Experiments and Discussion

5.1. NIR-VIS 2.0

The CASIA NIR-VIS 2.0 database [19] is the largest publicly available NIR HFR dataset. It contains 17,580 total images of 725 subjects. The dataset contains two views: View1 for algorithm development and parameter tuning, and View2 for performance reporting. View1 contains separate testing (probe/gallery) and training sets with

NIR-VIS 2.0	Rank 1	Std. Dev.	FAR=.001
IDNet	58.6	2.0	39.0
CDFL[9]	71.5	1.4	55.1
LMCFL[10]	75.7	2.5	55.9
[11]	78.5	1.67	85.8
C-CBFD[20]	81.8	2.3	47.3
[32]	86.2	0.98	81.29
Our Method	**87.1**	0.88	74.5

Table 2: Performance 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 [15]	87.8	98.2	95.8
C-DFD[17]	92.2	85.6	65.5
THFM [35]	99.28	99.66	98.42
[32]	99.38	-	92.25
Our Method	97.58	96.9	85.0

Table 3: Performance on View2 of CASIA HFB Face Database

NIR-VIS 2.0 View1 Identification Rate		Floating NIRNet Sections								
		1	1-2	1-3	1-4	All	2-5	3-5	4-5	5
Floating VisNet Sections	1	71.8	81.4	86.5	87.8	87.8	87.9	87.4	85.8	83.4
	1-2	80.7	81.5	85.7	87.6	87.5	87.7	87.6	85.6	82.3
	1-3	86.2	85.7	83.4	82.5	83.0	83.1	81.8	79.9	78.9
	1-4	86.1	86.7	82.0	77.0	73.5	73.7	72.6	70.7	70.2
	All	85.8	86.8	82.0	72.5	61.1	61.2	60.1	58.8	60.9
	2-5	85.7	86.6	81.7	73.3	61.2	61.2	59.7	58.6	60.9
	3-5	86.0	86.2	80.9	72.0	60.3	60.7	59.8	58.6	60.6
	4-5	85.5	86.0	80.4	71.0	61.1	61.1	59.7	60.9	60.0
	5	84.9	85.5	82.3	71.5	62.0	61.3	62.2	59.8	60.1

Table 4: View1 performance variation based on which sections of NIRNet and VisNet are altered during training. The rest of the sections are fixed to their initial IDNet values.

different subjects. The training set has 2,480 visible images and 6,270 NIR images of 357 subjects. The testing set has 6123 NIR probe images of the remaining 358 subjects and one visible gallery image per subject. View2 has 10 sub-experiments each of which has the same setup as View1 with slightly different numbers of images. NIR-VIS 2.0 is by far the most difficult NIR HFR dataset. In particular, the combination of pose variation, large gallery size, and single gallery image per subject make the dataset challenging. We view this dataset as the most relevant to real-world scenarios due to these challenges. Table 2 shows our method outperforms all others on View2 of this dataset. Additionally, the ROC curve on View2 is shown in Figure 4.

5.2. HFB

The CASIA HFB dataset [18] is an older and thus more widely used NIR HFR dataset. It has 5,098 total images from 200 subjects. It has a similar protocol setup to NIR-VIS 2.0 (View1 for tuning parameters, ten View2 experiments for reporting results). View1 (and each experiment in View2) split the dataset into 100 training subjects and 100 testing subjects. In View1, the training set contains 1,036 visible images and 1,438 NIR images, and the testing set contains 1,059 visible gallery images and 1,542 NIR probe images. The experiment protocols in View2 have similar statistics. HFB is less challenging than NIR-VIS 2.0 because there are fewer gallery subjects and multiple gallery images per subject. Additionally, there is significantly less pose variation in HFB as there is in NIR-VIS 2.0. These factors make HFB less relevant to real world applications. It's worth noting one aspect of HFB that makes it more difficult to train with: the relatively small number of training subjects. Models are more likely to fit to those specific subjects rather than generalizing. Our results on View2 of HFB and those of other methods are shown in Table 3 and our ROC curve is shown in Figure 5.

5.3. Layer Fixing Cross-validation

Table 4 shows the results of a cross-validation experiment used to choose which VisNet and NIRNet layers to alter during training. In addition to useful information for parameter selection, it also provides insight into the strengths and weaknesses of learning local and global features (see Table 5 for the localness of the features output by different network layers). The configurations that train only local features (the four in the upper left corner) perform significantly worse than those that consider global features. This supports our claim that it is important to learn features with a deep, global model. In the same vein, the configurations that produce the best results almost all allow learning in section 4 or section 5 in either VisNet or NIRNet. This shows that learning global features with a deep model can improve HFR performance. Additionally, the bottom right quadrant of the table shows the difficulty in learning global features and why they are generally not used in HFR. Any configuration where both nets have one of their two highest level sections (4 and 5) floating causes the network to overfit and perform poorly. This is how any deep approach to HFR would perform given the limited training data. We avoid this problem by fixing the high-level layers of one of the network halves to the strong initialization, yielding the performance shown in the upper right and bottom left quadrants. Without the IDNet initialization (i.e. with random initialization), we would have to train every layer of both VisNet and NIRNet, which yields an identification rate of 61.1 percent.

Another point that can be inferred from Table 4 is the relative importance of altering VisNet versus altering NIRNet. While the difference is usually small, it is generally better to alter more layers of NIRNet and leave more layers of VisNet fixed. This can be seen by comparing the identification rate of a combination with the that of the combination with swapped NIRNet and VisNet floating sections (swapping the row and column values on the table). For

Layer	Region Size
Input	1×1
pool1	6×6
pool2	16×16
pool3	36×36
pool4	76×76
pool5	100×100

Table 5: Sizes of image regions that affect features in a given layer. Each feature in a layer in the first column encapsulates visual information from an image region of corresponding size from the second column.

example, the performance when allowing all NIRNet sections and VisNet sections 1-2 to float is 87.5 percent, while the swapped version (NIRNet 1-2 and all sections of VisNet float) correctly identifies at a slightly lower rate (86.8 percent). Intuitively this makes sense as IDNet was initially trained to extract discriminative features from visible images and thus does not need to be tweaked as much when being used for that purpose.

5.4. Gallery Size

While the NIR-VIS 2.0 and HFB databases provide excellent benchmarks for NIR HFR, they do not cover all real-world scenarios. For example, it is possible to have a gallery of thousands or more subjects. In this section, we examine how the performance degrades with increasing gallery size. To accomplish this task, we append visible face images (one per subject) from the WebFace dataset used to train IDNet to the NIR-VIS 2.0 galleries. This provides a simulation of a situation with significantly more subjects. The results in Figure 6 show that our method is fairly robust to the increased gallery size.

6. Conclusion

In conclusion, we have presented a novel approach to NIR HFR. We have proposed to use convolutional neural networks to learn deep, global features that capture discriminative information in NIR and visible face images. Moreover, we coupled the networks so they produce domain-independent features that can be compared directly. We prevented overfitting by initialization with a visible face identification network trained on a very large visible face dataset. We evaluated our approach on two benchmark databases and additionally investigated how our method scales with larger gallery sizes.

7. Acknowledgements

All plots in this work were generated with Matplotlib [7]

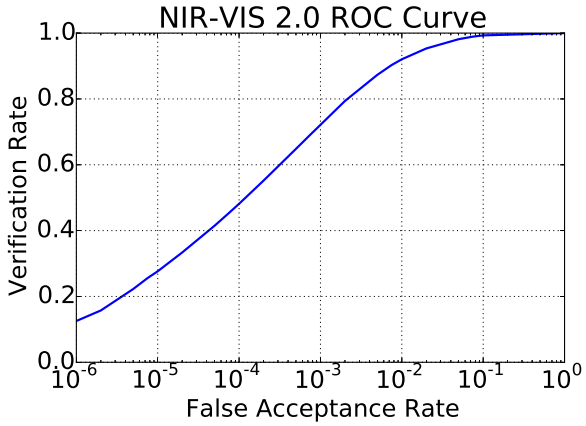

Figure 4: ROC Curve for CASIA NIR-VIS 2.0 dataset.

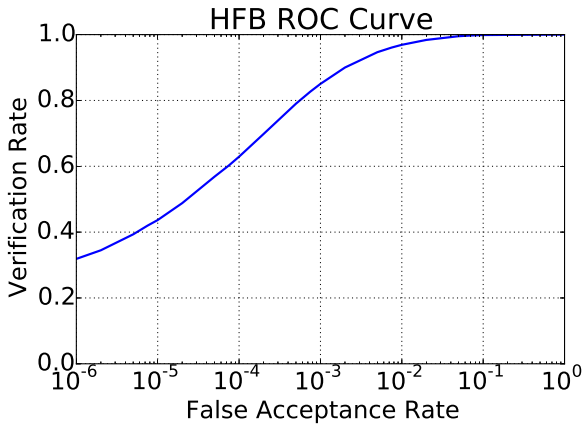

Figure 5: ROC Curve for CASIA HFB dataset.

References

[1] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Computer Vision–ECCV 2012*, pages 566–579. Springer, 2012.

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[3] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.

[4] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society con-*

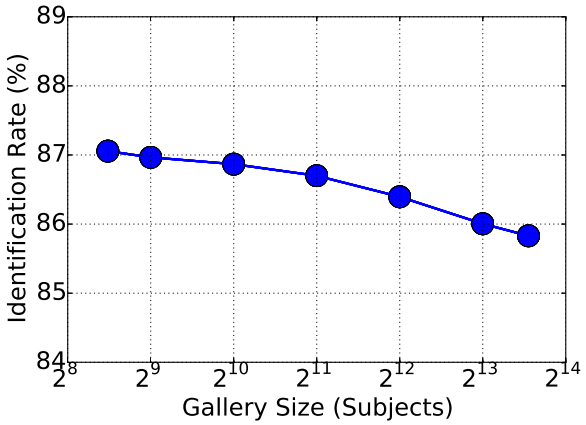

Figure 6: Performance degradation on NIR-VIS 2.0 View2 with increased gallery size.

ference on, volume 2, pages 1735–1742. IEEE, 2006.

[5] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[6] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] B. Klare and A. Jain. Heterogeneous face recognition: Matching nir to visible light images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1513–1516, Aug 2010.

[15] 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.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[17] 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.

[18] S. Z. Li, Z. Lei, and M. Ao. The HFB face database for heterogeneous face biometrics research. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2009.

[19] S. Z. Li, D. Yi, Z. Lei, and S. Liao. The casia nir-vis 2.0 face database. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 348–353. IEEE, 2013.

[20] 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.

[21] S. OuYang, T. M. Hospedales, Y. Song, and X. Li. A survey on heterogeneous face recognition: Sketch, infra-red, 3d and low-resolution. *CoRR*, abs/1409.5114, 2014.

[22] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *Proceedings of the British Machine Vision*, 2015.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 815–823, June 2015.

[24] K. Simonyan, O. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.

[25] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[26] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[27] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1891–1898, June 2014.

[28] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2892–2900, June 2015.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.

Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[30] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708, June 2014.

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2746–2754, June 2015.

[32] 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.

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

[34] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[35] 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.