Intelligent Synthesis Driven Model Calibration: Framework and Face Recognition Application

Jordan Hashemi
Duke University
jordan.hashemi@duke.edu

Qiang Qiu
Duke University

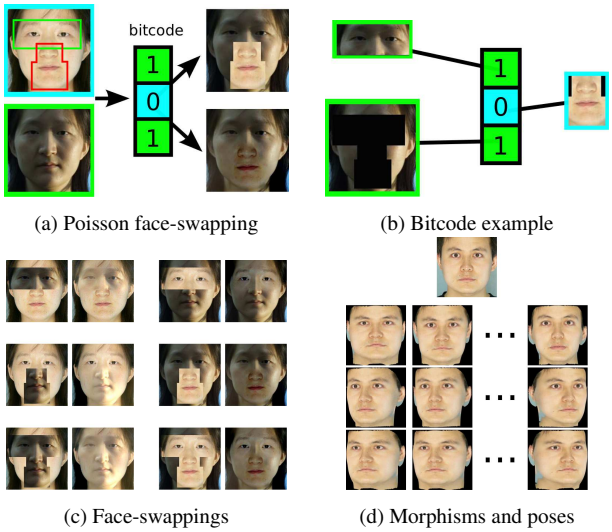
Guillermo Sapiro
Duke University

Abstract

Deep Neural Networks (DNNs) that achieve state-of-the-art results are still prone to suffer performance degradation when deployed in many real-world scenarios due to shifts between the training and deployment domains. Limited data from a given setting can be enriched through synthesis, then used to calibrate a pre-trained DNN to improve the performance in the setting. Most enrichment approaches try to generate as much data as possible; however, this blind approach is computationally expensive and can lead to generating redundant data. Contrary to this, we develop synthesis, here exemplified for faces, methods and propose information-driven approaches to exploit and optimally select face synthesis types both at training and testing. We show that our approaches, without re-designing a new DNN, lead to more efficient training and improved performance. We demonstrate the effectiveness of our approaches by calibrating a state-of-the-art DNN to two challenging face recognition datasets.

1. Introduction

Recent advances in DNNs have greatly impacted the face recognition community. DNNs that have achieved state-of-the-art performances on benchmark datasets trained using millions and hundreds of millions of training images [15, 23, 25]. With this said, they are still prone to suffer severe performance issues when deployed in many real-world scenarios. These performance issues stem from shifts between the training and deployment domains such as image resolution, lighting conditions, occlusions, ethnicities of subjects, among others. For example, consider the deployment scenario with a face recognition system at a remote checkpoint location, where the task is to identify images of subjects captured on a mobile phone. Following most state-of-the-art methods, in particular those used by systems without access to the hundreds of millions of private collections, the training set will be made up of labelled images taken from the internet (usually of celebrities with makeup and at ideal lighting conditions). While in this deployment scenario, the testing set contains only images taken from the user's mobile phone, possibly at poor lightning conditions and reduced image quality.



(a) Poisson face-swapping (b) Bitcode example

(c) Face-swappings (d) Morphisms and poses

Figure 1: Examples of face synthesis methods. (a) shows outline of face-swapping approaches. Given the input images on the left, our method performs Poisson face swapping (lower right image in (a)) compared to direct cloning (upper right in (a)). (b) shows a bitcode example with provided face regions. (c) demonstrates face-swapping results where the odd numbered columns are results from cloning face regions and the even columns show results from our proposed Poisson cloning method using Poisson image editing. (d) demonstrates face morphisms and pose variations where the top row shows the base image, the second row shows face morphism results, the third row shows pose variation results, and the last row shows results from combining variations of morphisms and poses.

One straightforward way to address this scenario is to collect a sufficient amount of labelled data in the expected deployment setting; however, this is an intensive and tedious solution and is often times impractical. A more practical solution is to obtain a limited set of labelled data in the

deployment setting and then use it to calibrate (adapt) a pre-trained model. A major concern with using limited training data is overfitting. Fortunately, with advances in computer vision and graphics, it is possible to enrich a limited dataset with face synthesis methods where one uses the provided annotated face images to generate new ones [4, 13]. For face recognition, intra-class face synthesis (detailed in Section 2) is especially powerful since it enriches the training dataset while at the same time retaining the original class labels. These synthesis approaches can also be exploited at testing time to generate more realizations of testing images.

Although face synthesis approaches have been shown to be powerful methods to enrich limited datasets, the current driving force behind synthesizing is to generate as much data as possible to feed into a DNN. Performing face synthesis in such a blind manner is not only very time consuming and computationally expensive during training, but can be prone to generating redundant data that will have minimal impact or generating data that is too extreme which will harm the calibration method. Smart synthesis during deployment (testing) is even more critical.

In this work, we explore the effectiveness of face synthesis for calibrating a pre-trained DNN to multiple constrained deployment scenarios where limited training data is available. To achieve this we develop face synthesis methods and propose information-driven approaches to exploit and optimally select face synthesis types and samples. We show that our approaches lead to more efficient training and improved testing performances.

Our main contributions are:

- We develop a Poisson face-swapping method and person-specific 3D morphable model to synthesize novel face images.
- We propose information-driven approaches to guide (face) synthesis during training and testing.
- We demonstrate how to efficiently use face synthesis to calibrate a DNN given multiple constrained deployment scenarios, thereby demonstrating how DNNs can become even more powerful than what they were designed/trained for.

We should mention that while exemplified with the important task of face recognition, the approach here proposed is general and illustrates the importance of intelligent synthesis. Moreover, the pipeline steps like the actual synthesis models could be replaced by others such as generative adversarial networks (GANs), potentially further improving the results and without reducing the relevance of the key concepts here introduced.

1.1. Related work

Enriching datasets with synthesized data has been explored in many visual recognition tasks including body pose estimation [21], text localization [3], gaze estimation [24], and face recognition [4, 13]. Notably, [4] employ face-swapping to clone combinations of facial regions, enriching face datasets by generating both novel intra-class and inter-class (new subjects) face images. The work in [13] fits a generic 3D morphable model (3DMM) to face images and enriches face datasets by synthesizing with different shape, pose, and mouth variations. At testing they perform synthesis on the test images but only to create pose variations. These works focused on training DNNs from scratch, without any intelligent selection of the synthesis, and demonstrated that these face synthesis methods improved performance. We will exploit these synthesis methods.

The work in [16] explored the effectiveness of simple image transformations (scaling, mirroring, flipping, etc.) for the task of image classification. They demonstrated the importance of iteratively selecting transformation types that are most informative during training to increase efficiency. In addition, they showed the importance of utilizing informative transformations at both training and testing. At each iteration for selecting a transformation type, their approach trains classification models for every possible transformation type, then picking the type that provides the greatest increase in performance. Training models for every transformation type at each iteration can be computationally expensive, and does not easily allow for combinations of transformations to be considered.

Finally, and following in part the recent advances in GANs (see for example [24]), multiple synthesis techniques have been proposed in literature. While ours are more tailored and carefully designed to the challenges in face recognition (needing less data and having more control of attributes), those techniques could be incorporated as part of the proposed intelligent synthesis framework.

2. Face synthesis methods

Given a limited training dataset from the target domain, the goal is to be able to enrich the dataset while preserving the annotated class labels of the target domain; in this section we outline three such face synthesis methods. First, we describe our Poisson face swapping method that couples face-swapping with Poisson editing to synthesize realistic face images. In addition, we use a 3DMM to generate novel realizations and viewpoints of a face image through deforming a fitted person-specific face model and capturing realizations at different head poses. We preprocess each face image by first detecting facial landmarks [8], and then using 7 landmarks locations (namely the two right eye corners, two left eye corners, nose tip, and the two mouth corners) we align the face to a canonical frontal face model through a similarly transform. After alignment, we finally combine the two methods.

Images provided by the target dataset are referred to

as base images. $\mathrm{D}\{n, m\}$ represents a base image from subject n and image number m, where $n \in N$ and $m \in M$. $\mathrm{S}(\mathrm{D}\{n, m\})$ represents a set of synthesized face images generated from the base image $\mathrm{D}\{n, m\}$, where $|\mathrm{S}(\mathrm{D}\{n, m\})|$ is the total number of face synthesis types, and each synthesis type $i \in |\mathrm{S}(\mathrm{D}\{n, m\})|$ is represented by $s_i(\mathrm{D}\{n, m\})$.

2.1. Poisson face-swapping

Given any two face images, we synthesize new face images through combinations of their face regions. Motivated by [4], we use automatically detected facial landmarks to define 3 face regions: eyes, nose and mouth, and rest of the face (see Figure 1). To synthesize a new face image, we define the triplet (b, d, c) where b and b correspond to two images that will be mixed, and the bitcode $c \in \{0, 1\}^3$ defines which face regions will be taken from each image. A zero in the bitcode c represents the corresponding face region will be taken from image b, whereas a one represents the corresponding face region will be taken from image d.

Just swapping the face regions introduces unnatural artifacts, including major image gradients around the swapped face regions and drastic contrast differences between the swapped regions. More realistic synthesized face images can be generated by viewing the process of face swapping as an application of guided interpolation that can be solved via Poisson image editing methods [2, 18]. Guided interpolation aims at seamlessly cloning novel objects or image sections into a background image. In this case of face swapping, the objective is to seamlessly clone face regions of one face image over to another face image. The background image is created based on the bitcode c and is defined as the image created from the combination of the 'rest of the face' region and other face regions that share the same code entry as the 'rest' entry. Let $\mathcal{B}$ be the background image and $\Omega \in \mathcal{B}$ be the domain of the face regions which we wish to replace, where $\partial\Omega$ is the boundary of these face regions. The known image values in $\mathcal{B}$ are denoted as f^*, while f are the unknown image values defined over the interior of Ω. Furthermore, let g^* be the known image values of the face regions we wish to clone onto the face regions in the background image, where its gradients ∇g^* are used as the guidance vector field. The goal is to minimize the difference of the gradient vector fields between the background image and the desired face regions we wish to clone,

$$\min_f \int_\Omega \| \nabla f - \nabla g^* \|^2, \quad s.t.\ f|_{\partial\Omega} = f^*|_{\partial\Omega}, \tag{1}$$

whose solution is the Poisson Equation over the domain Ω with imposed Dirchlet boundary conditions,

$$\nabla^2 f = \nabla^2 g \text{ over } \Omega, \text{ and } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \tag{2}$$

where $\nabla^2 = [\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}]$. Many approaches can be used to solve Equation (2), for this work we use the *finite difference method* implementation [2, 18]. Thus the final synthesized image will contain computed pixels f inside the swapped facial regions defined by Ω, and background pixel values f^* outside Ω. Figure 1 illustrates examples of our proposed face-swapping method, which is both very simple (based on swapping) and very realistic (thanks to Poisson editing).

2.2. Person-specific face morphisms and poses

To generate novel morphisms of any face image, we first iteratively morph a 3DMM so that corresponding landmarks between the face image and 3DMM align with one another. We utilize the Basel Face Model [17], a linear principal component analysis 3DMM parameterized by 199 shape principal components. It is a very dense model consisting of q = 53,490 depth vertices and 160,470 faces. From a 3DMM new 3D face models, $\mathbf{BFM}(\alpha) \in \mathbb{R}^{3q}$, are synthesized via

$$\mathbf{BFM}(\alpha) \quad = \quad \mu + \mathbf{U}\alpha\,, \tag{3}$$

where $\mu \in \mathbb{R}^{3q}$ is the mean face shape, $\alpha \in \mathbb{R}^{199}$ contains the shape parameters, and $\mathbf{U} \in \mathbb{R}^{3q \times 199}$ is the shape basis. Note that in (3) $\mathbf{BFM}$ and μ contain q depth vertices in a concatenated form, therefore are written as $3q$ vectors.

For learning a person-specific face model from an input face image, the goal is to determine the optimal α values that correctly register the BFM model to the input face image. Let us introduce two rigid transformation parameters, a scale and rotation parameter $\mathbf{R}$ and a translation parameter $\mathbf{t}$. Then the following two term loss function is minimized:

$$E(\theta) \quad = \quad E_l(\theta) + \eta E_s(\theta), \tag{4}$$

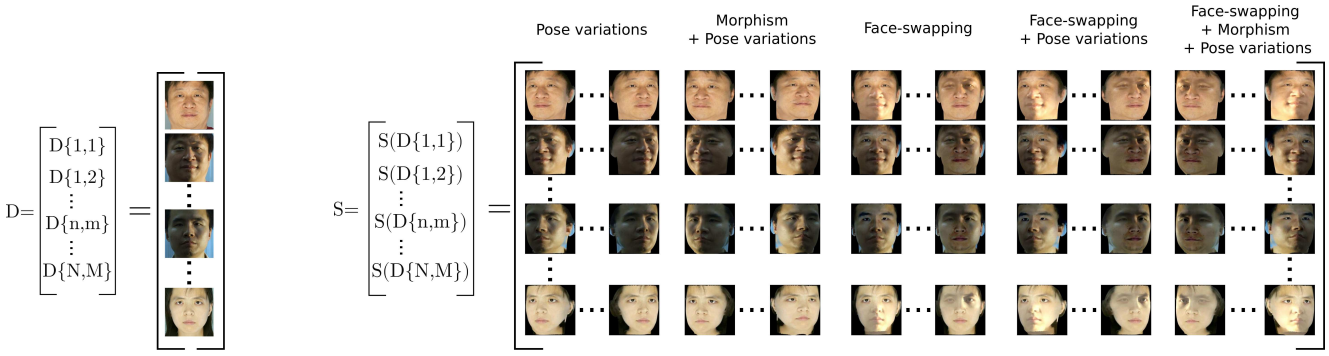
where $\theta = \{\alpha, \mathbf{R}, \mathbf{t}\}$ contains the shape and rigid transform parameters. The two terms represent a landmark term E_l and a regularization term E_s, where the regularization term is weighted by the stiffness parameter η. More specifically these terms are given by

$$E_l(\theta) \quad = \quad \sum_{(p,l)\in L} \|\mathbf{R}\,(\mu_p + \mathbf{U}_p\alpha) + \mathbf{t} - l\|^2, \tag{5}$$

$$E_s(\theta) \quad = \quad \|\alpha\|^2, \tag{6}$$

where μ_p and $\mathbf{U}_p$ represent the rows corresponding to vertex p in μ and $\mathbf{U}$ respectively. The term (5) minimizes the distances between the corresponding BFM and image landmarks L. The regularization term (6) enforces small values for the shape parameters α and is guided by the stiffness parameter η. The proposed method is an adaptation of the method in [14], where the authors focused on fitting a 3DMM to an input face mesh.

The fitting defined by (4) is an iterative process where the stiffness parameter η is increased after each iteration.



(a) Collection of base images

(b) Collection of synthesized images

Figure 2: Organization of collections for base and synthesized face images. For any image m from subject n, $\mathrm{D}\{n, m\}$ represents the base face image and $\mathrm{S}(D\{n, m\})$ represents the set of synthesized face images. Face synthesis types, s_i, span the column space of S.

As η increases, it restricts the amount the 3DMM can deform. Convergence is achieved when the parameter set differs by less than a small margin between iterations. Once the optimal face model $\mathbf{BFM}^*$ has converged, each vertex is assigned a texture index by directly sampling from the nearest pixel location on the input image.

Since the learned person-specific face model $\mathbf{BFM}^*$ is derived from a 3DMM, new face models $\mathbf{Z}(\tilde{\alpha})$ based on the input face image can by synthesized by varying the shape parameter $\tilde{\alpha}$,

$$\mathbf{Z}(\tilde{\alpha}) \quad = \quad \mathbf{BFM}^* + \mathbf{U}\tilde{\alpha}\,. \tag{7}$$

Notice that (7) and (3) are identical except the mean face shape in (7) is replaced by the learned person-specific face model. Since $\mathbf{Z}$ and $\mathbf{BFM}^*$ have one-to-one vertex and face correspondences, the texture from the learned person-specific model can be directly transferred to the synthesized face model $\mathbf{Z}$.

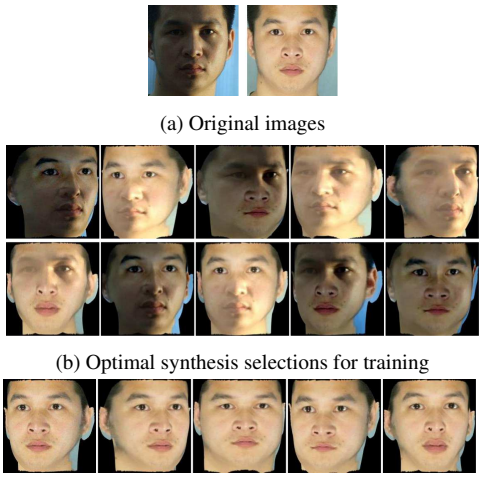
We can generate novel poses of any 3DMM by rendering at different viewpoints. We generate poses and morphisms similar to those in [13], where the authors demonstrated that these morphisms and poses increase performance for face recognition (Figure 1d).

2.3. Combinations of methods

Face synthesis can be performed through combinations of the methods described above. For the work presented here we focus on intra-class synthesis, where the synthesized face images are always assigned a class label belonging to the known training classes. We refer the reader to [4] for work where face-swapping was performed to create unseen labels.

3. Information driven synthesis for calibration

Face synthesis is known to provide meaningful information for the task of face recognition[13]; however, the process of synthesizing face images is very time consuming, requires large storage space, and can bias training. Often, many of these synthesized faces contain redundant information across the other synthesized faces and the base images they are produced from. We propose two information theoretic driven approaches to make face synthesis more efficient for training and increase performance at testing.



(a) Original images

(b) Optimal synthesis selections for training

(c) Optimal synthesis selections for testing

Figure 3: Visual results for optimal synthesis selections for training and testing. (a) Images provided by the dataset. The rows in (b) show results for optimal selection of synthesis types for training guided by ME (top) and MMI (bottom). ME favors synthesis at extreme poses and morphisms, whereas MMI chooses a more balanced synthesis subset both in training and in testing. The columns in (c) shows synthesis selections guided by MMI for testing from the right-most base images in (a).

3.1. Modeling face synthesis as a Gaussian Process

Let the features from a face image generated from synthesis type i be represented by L_2 normalized d-dimensional vector $s_i(\mathrm{D}\{n, m\}) \in \mathbb{R}^d$, where a synthesis type is defined as any combination of the face synthesis methods in Section 2. Features from synthesized face im-

ages with redundant information will have high similarities with one another. We model the face images as a Gaussian Process (GP) which is represented by a mean function and a positive-definite kernel function $\mathcal{K}$. Furthermore, we combine all synthesis outputs across the subjects to create a collection S of face images from different types of synthesis where each column $s_i \in \mathbb{R}^{dMN}$ represents a different synthesis type and the rows are organized in subject-specific blocks (Figure 2b). From S, we define the kernel function across each pair (i, j) of synthesis types as $\mathcal{K}_{s_i,s_j} = sim(s_i, s_j)$, where $(s_i, s_j) \in |\,\mathrm{S}\,|$ and $sim(\cdot)$ is the cosine similarity. A GP allows us to model any synthesis type as a Gaussian distribution whose conditional variance is given by $\sigma_{s_i|\,\mathrm{S}} = \mathcal{K}_{s_i,s_j} - \mathcal{K}_{s_i,\mathrm{S}}\,\mathcal{K}^{-1}_{\mathrm{S,S}}\,\mathcal{K}_{\mathrm{S},s_i}$ where $\mathcal{K}_{s_i,\mathrm{S}}$ is the covariance vector between s_i and S. For computational efficiency it is beneficial to use a kernel function with compact support. One way is to set a small threshold ϵ where one removes all synthesis types i for which $|\,\mathcal{K}_{s_i,s_j}\,| \leq \epsilon$.

Our objective is to select an optimal subset of synthesis types that are most representative to space of all possible types. A straightforward approach is to minimize the conditional entropy $H(\cdot)$ of the non-selected synthesis types given the already selected subset S^*:

$$\arg\min_{\mathrm{S}^*} H(\mathrm{S} \setminus \mathrm{S}^* \,|\, \mathrm{S}^*) \Rightarrow \arg\max_{\mathrm{S}^*} H(\mathrm{S}^*). \tag{8}$$

In turn this selects the optimal subset with maximum entropy, and in practice it biases extreme synthesis types (Figure 3b), which is not ideal. These shortcomings of optimal selection using *entropy criterion* have also been observed in [9, 19] for tasks related to sensor placement and action attribute learning. We will address this problem next.

3.2. Optimal synthesis selection for training

To diminish the bias of selecting extreme synthesis types, we add the constraint of selecting synthesis types that are most representative of all the non-selected types. Specifically, we want

$$\begin{aligned} &\arg\max_{\mathrm{S}^*} H(\mathrm{S} \setminus \mathrm{S}^*) - H(\mathrm{S} \setminus \mathrm{S}^* \,|\, \mathrm{S}^*) \\ &\Rightarrow \arg\max_{\mathrm{S}^*} \mathcal{I}(\mathrm{S}^*; \mathrm{S} \setminus \mathrm{S}^*), \end{aligned} \tag{9}$$

which is equivalent to selecting the subset that maximizes the mutual information $\mathcal{I}(\cdot)$ between the selected types S^* and the non-selected types $\mathrm{S} \setminus \mathrm{S}^*$.

Selecting the optimal subset can be done with a greedy algorithm as follows. Initialize $\mathrm{S}^* = \emptyset$. Then iteratively choose the next best synthesis type y^* from $\mathrm{S} \setminus \mathrm{S}^*$ that provides the maximum increase in mutual information,

$$\begin{aligned} &\arg\max_{y^* \in \mathrm{S} \setminus \mathrm{S}^*} \mathcal{I}\,(\mathrm{S}^* \cup y^*; \mathrm{S} \setminus (\mathrm{S}^* \cup y^*)) - \mathcal{I}(\mathrm{S}^*; \mathrm{S} \setminus y^*) \\ &\Rightarrow \arg\max_{y^* \in \mathrm{S} \setminus \mathrm{S}^*} H(y^* \,|\, \mathrm{S}^*) - H(y^* | \bar{\mathrm{S}^*}), \end{aligned} \tag{10}$$

where $\bar{\mathrm{S}^*}$ denotes $\mathrm{S} \setminus (\mathrm{S}^* \cup y^*)$. Since the conditional entropies are from a Gaussian random variable, they have the closed form solution

$$H(y^* \,|\, \mathrm{S}^*) = \frac{1}{2} \log(2\pi e\, \sigma^2_{y^* \,|\, \mathrm{S}^*}). \tag{11}$$

This greedy approximation algorithm can be solved in polynomial-time [9].

So far we have focused on a collection of synthesized face images only; however, the optimal selection of k_{train} synthesized face images can be guided by incorporating the provided base face images of the subjects, D, simply initializing $\mathrm{S}^* = \mathrm{D}$. Then we iteratively choose the next best synthesis type according to (10), until $|\,\mathrm{S}^*\,| = k_{train}$.

3.3. Optimal synthesis selection for testing

In the previous section we were concerned with determining the types of face synthesis that provide the most information with respect to all of the training data in order to calibrate (adapt) the model to the target domain. At testing time we wish to further exploit the synthesis types that were performed in training to improve testing performance; however, the trade-off between accuracy and computation is crucial, thus using synthesis methods with redundant information is not desirable. Selecting the synthesis types that are most similar to each other will lead to selecting a local group of synthesized face images with redundant information. Instead we want to select synthesis types that are similar to each other but are also informative with respect to types that have already been selected. Again, we can use the same maximum mutual information approach (9), but now further restrict the selections to those that were employed during training. Thus the algorithm for picking k_{test} optimal synthesis types for testing is to first initialize $\mathrm{S}^* = \mathrm{D}$. Then iteratively choose the next best synthesis type y^* from $\mathrm{S} \setminus \mathrm{S}^*$ according to (10) until $|\,\mathrm{S}^*\,| = k_{test}$. At testing it is not guaranteed that multiple of images per subject will be available, thus for testing we restrict the face synthesis collection to be defined by face morphism and pose variations only. Selecting the optimal synthesis types for testing is very efficient since it does not require any additional synthesis to be performed since it only considers synthesis types that are a subset of the synthesis types that were performed during training.

3.4. Model calibration

To calibrate the deep features to a given deployment scenario, we first normalize the deep features via L_2 normalization, then freeze the network except for a newly added fully connected layer. This added layer serves as an embedding to map the deep features to the given deployment scenario. There are many different embedding constraints that can be deployed, such as pair-wise[1] or low-rank[20].

For this work we use the triplet-loss [23, 26]. Let $x(\mathbf{I})$ denote the L_2-normalized feature representation of face image $\mathbf{I}$, where $\|x(\mathbf{I})\| = 1$. The triplet loss handles a triplet of examples, namely $\{x(\mathbf{I}^a), x(\mathbf{I}^+), x(\mathbf{I}^-)\}$, where $x(\mathbf{I}^+)$ is the positive feature representation sharing the same class as the anchor $x(\mathbf{I}^a)$, whereas $x(\mathbf{I}^-)$ is the negative representation belonging to a different class. The goal is to learn the embedding W, so that the embedded feature space $\phi(x) = Wx$ minimizes the distance between the anchor-positive pairs while maximizing the distance between the anchor-negative pairs. For notation sake, the sub and superscripts of $\mathbf{I}$ will be transferred to x; so $\mathbf{x}$ represents the feature of any image input $\mathbf{I}$, and furthermore, $\mathbf{x}^a$, $\mathbf{x}^+$, and $\mathbf{x}^-$ represent the normalized feature representations of anchor, positive, and negative images respectively. The triplet loss to be minimized is as follows:

$$E_{triplet}\left(\phi(\mathbf{x}^a), \phi(\mathbf{x}^+), \phi(\mathbf{x}^-)\right) = \max\{0, \gamma + \|\phi(\mathbf{x}^a) - \phi(\mathbf{x}^+)\| - \|\phi(\mathbf{x}^a) - \phi(\mathbf{x}^-)\|\}.$$

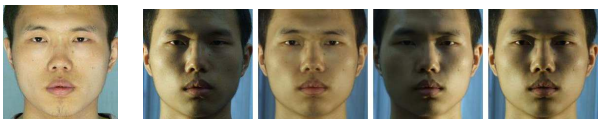
Here γ acts as a margin parameter between the negative and positive pairs, The task of choosing a triplet is crucial. Similar to [15, 23], we perform hard-negative mining for triplet selection. An epoch considers all anchor-positive pairs in the training set. We use an initial learning rate of 0.1 and convergence is achieved when the embedding differs by a small margin between epochs.

4. Experimental validation

We use the state-of-the-art *VGG-Face* [15] DNN to extract feature representations of our face image. To date, *VGG-Face*[2] is the highest performing publicly available model for facial recognition on the gold-standard *Labeled Faces in the Wild* [5] dataset, achieving $\mathbf{98.95}\%$ verification accuracy. It is a 16 layer DNN trained on the *VGG Face Dataset* [15] which contains 2.6 million images taken from the web of over 2,000 celebrities. A driving component of *VGG-Face*'s success is due to the access of millions of images taken from the web during training, thus sharing and capturing the same domain as many other validation datasets. On the YouTube Faces [27] benchmark it also achieves near-perfect accuracy: $\mathbf{97.30}\%$. To validate our proposed calibration approaches, we chose two datasets that are from constrained domains, and where *VGG-Face* performs less than optimal on. Namely we chose the *OFD*[1] and CASIA NIR-VIS 2.0 [11] datasets, where out-of-the-box *VGG-Face* achieved $\mathbf{80.70}\%$ and $\mathbf{67.47}\%$ rank-1 classification scores respectively. The *OFD* dataset contains prominent illumination challenges, whereas the CASIA NIR-VIS 2.0 dataset contains images from different modalities.

[2] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

[1] http://gr.xjtu.edu.cn/web/jianyi/tt



(a) Gallery (b) Probes

Figure 4: *OFD* gallery and probe base images from from a given subject for testing. Only the gallery and first probe image types (two left-most images) are provided during training.

Since the last layer of *VGG-Face* is trained on labels from the *VGG Face Dataset*, we discard it and use the second to last layer as our deep feature representation. We perform principal component analysis to reduce the deep feature representation to 512 dimensions. We then learn a 512-by-512 embedding W via the triplet-loss to adapt the model to the different dataset domains. Lastly, we use the cosine similarity score to perform matching. If synthesis is performed during testing, the average matching score across all synthesized images from a given testing image is used. Note that the method here proposed will enjoy this state-of-the-art method without having to re-design it when adapting it to new domains.

It takes our system, Intel Core i7 5820K computer with 64GB DDR4 RAM and an NVIDIA GeForce Titan X, 800 milliseconds to synthesize new faces.

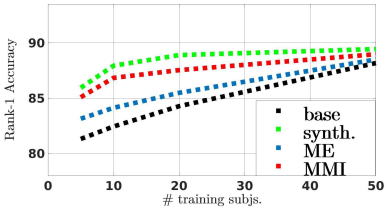
4.1. Results on *OFD*

OFD is a Chinese face dataset containing 33,669 images across 1,247 subjects. Images from this dataset were taken in a controlled setting, where poses, lighting conditions, background color, and facial accessories were controlled, making it an ideal dataset to demonstrate effectiveness of calibrating a trained model to a new deployment scenario. Without any calibration, *VGG-Face* performs with an 80.70% rank-1 classification score in accordance with our validation procedure explained next.

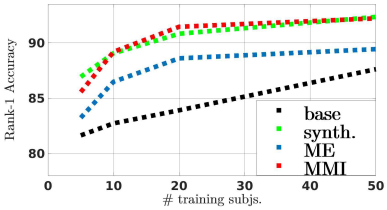
For all results presented we conduct 5-fold cross-validation. Thus, we first separate the subjects into 5 validation partitions (around 250 subjects per partition). For each validation partition, we test with only the subjects in a given partition and train with subjects in the remaining partitions.We compute classification scores from the data in the testing partition using a gallery depth of 1 and probe depth of 4 per subject (see Figure 4). We focus our experiments around scenarios where limited training data is provided: $M = 2$ images per subject and $N = \{5, 10, 20, 50\}$ subjects are provided. Thus we further divide each training partition into subsets with non-overlapping training subjects. For a given validation partition, classification results are computed by averaging the testing partition's scores across all the subsets in the respective training partition. Then the final results shown are the average scores throughout the 5 validation partitions. We also explore the effects of k_{train}

Amount of training images			
N	base	synth.	ME / MMI
5	10	1,410	60
10	20	2,820	120
20	40	5,640	240
50	100	14,100	600

(a) Amount of deployment data



(b) testing: base



(c) testing: SMMI

Figure 5: Rank-1 classification performance as the number of training subjects, N, increases. Initial rank-1 score for un-calibrated model is 80.70%. Our proposed training and testing approaches incorporating MMI, require less than 5% of the total amount of synthesized data at training and achieve highest performance. It also outperforms using ME to select optimal synthesis. For these experiments $k_{train} = 10$ and $k_{test} = 5$.

and k_{test}, which define the size of the training and testing synthesis subsets respectively.

Incorporating synthesis at training and testing. For each subject provided in the training set, we generate a total of 280 synthesized face images defined by a combination of 8 face-swappings, 5 morphisms, and 7 pose variations. Examples of the synthesized face images for a given subject can be seen in the first two rows of Figure 2b. We compare our proposed optimal synthesis selection guided by *maximum mutual information* for training approach (MMI) (9) to three other training approaches, namely only using the provided base face images (**base**), using the base and all synthesized face images (synth.), and optimal synthesis selection guided by *maximum entropy* (ME) (8). We also compare our proposed optimal synthesis selection guided by *maximum mutual information* testing approach (testing: MMI), to cases when no synthesis is used at testing (testing: base).

Figure 3 shows visual results of optimal selections for both training and testing. Synthesis guided by ME tends to favor synthesis results at extreme poses and morphisms, whereas MMI selects a more balanced subset. In addition, MMI selects similar synthesis types at testing, and as we show below this leads to improved performance.

Optimal synthesis selection. Table 1 shows results when we vary the amount of synthesis types for training k_{train}. As expected, increasing the amount of synthesis at training leads to better performance (trends in rank-1 and rank-5 results for testing: base). In addition, incorporating synthesis at testing leads to greater increases in performance in all training setups. Furthermore, using our proposed MMI approach to select only 10 out of the 280 synthesis types at training, we achieved higher performance than using all 280 synthesis types.

As we observed in Table 1, highest performance is achieved when using synthesis guided by MMI during both training and at testing. Testing performance is sensitive to the amount of synthesis types k_{test} at testing. We observed the optimal k_{test} selection to be 5 and when we considered all possible synthesis types at testing, it slightly hindered performance (rank-1 classification scores were 89.80% and 86.38% for $k_{test} = 5$ and 35 respectively). We speculate this is possibly due to noise introduced from face synthesis, but considering computation costs and time at testing, lower values of k_{test} are preferred.

	testing: base		testing: MMI	
k_{train}	**ME**	**MMI**	**ME**	**MMI**
0 (**base**)	82.37	82.37	82.77	82.77
5	83.16	86.35	84.58	87.96
10	84.06	86.94	85.89	89.80
50	86.22	86.87	89.24	**90.13**
280 (synth.)	86.61	86.61	89.66	89.66

Table 1: *OFD* rank-1 classification scores across testing setups for varying training setups and synthesis selection parameter for training, k_{train}. Results shown for calibration experiment with $N = 10$ training subjects and $k_{test} = 5$. Note that $k_{train} = 0$ is the same as training only with base images while $k_{train} = 280$ means training with all synthesized images. By incorporating MMI during training and testing, we achieve highest results while only using a small subset of the total synthesis data.

Effectiveness of synthesis. Figure 5 fully demonstrates the effectiveness of using our proposed MMI approach during training and at testing. Performing any type of synthesis at training (Figure 5b) drastically improved rank-1 performance, where MMI and synth. showed the largest increase. Furthermore, the best results were achieved when MMI was also employed at testing (Figure 5c). Using MMI to select optimal synthesis types for training and testing, achieved nearly 93% rank-1 accuracy compared to 88% by using just the base images, and was able to efficiently use less than 5% of the total synthesis data, while achieving similar results to using all of the synthesis data (10 synthesis types vs. 280). This is further investigated in the Receiver Operating Characteristic (RoC) curve in Figure 6 and results are recorded in Table 2. Without touching or re-designing the state-of-the-art *VGG-Face* model, we were able to efficiently adapt it to the *OFD* dataset, drastically improving true positive rate (TRP) at 1% false alarm rate (FAR) from 0.79 to 0.88 and improve rank-1 classification from $\mathbf{80.70}\%$ to $\mathbf{89.19}\%$.

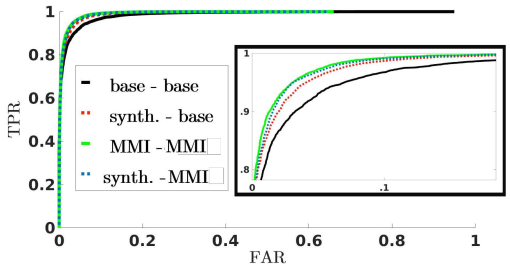


Figure 6: RoC curves and performance for calibration experiments where $N = 10$, $k_{train} = 10$, and $k_{test} = 5$. (a) RoC curves for 4 training and testing approaches. Employed training and testing methods are represented in the legend, and are separated by a '-'. Black solid line represents when only base images are used. The dotted lines represent cases when all synthesized images are used for training. The red line represents when synthesis was not performed at testing, while the blue is when MMI was employed at testing. Solid green line represents our proposed approach of using MMI both at training and at testing. The subplot shows a zoomed region of the RoC curves.

Training and testing setups		amount of training images	TPR@ 0.01FAR	rank-1 accuracy
k_{train}	k_{test}			
0 (base)	0 (base)	20	0.81	82.44
280 (synth.)	0 (base)	2,820	0.86	86.94
280 (synth.)	5 (MMI)	2,820	0.87	89.06
10 (MMI)	5 (MMI)	**120**	**0.88**	**89.19**
VGG-Face			0.79	80.70

Table 2: Results on calibrating the *VGG-Face* model to the *OFD* dataset. MMI guided calibration for training and testing improved the rank-1 performance for *VGG-Face* from 80.70% to 89.19% without altering the DNN model.

4.2. Results on CASIA NIR-VIS 2.0

The CASIA NIR-VIS 2.0 dataset [11] is the largest cross-spectrum face dataset available, containing 17,580 images across 725 subjects in both the near-infrared (NIR) and visual (VIS) spectrums. Benchmark results shown are rank-1 classification averages and standard deviations taken across from 10 validation sets where the NIR and VIS images are the probes and gallery respectively (Figure 7). The dataset contains images from two modalities (NIR and VIS) making this a very challenging dataset where *VGG-Face* achieves a 67.47% average rank-1 classification score. When enriching the training set, we treat each domain independently, in other words, we do not perform synthesis on images from different domains. Specifically, for every subject we first randomly sample $M = 3$ images in each modality. Across the M images we perform combinations of 6 Poisson face-swappings, 3 person-specific morphisms, and 3 pose variations. In total, we generate 324 different synthesis types for each subject. In the triplet embedding optimization, we also preserve domain information by requiring the anchor selection to be from a different domain than the positive and negative selections.

Figure 7: Examples of images and optimal synthesis types from CASIA NIR-VIS 2.0 dataset. First column shows gallery (top) and probe (bottom) images from the dataset. The remaining columns are synthesis results from the provided base images chosen at testing by MMI.

Lezama et al. [10][1] (2016)		89.59 ± 0.89
Yi et al. [28](2015)		86.16 ± 0.98
Saxena et al.[22] (2016)		85.90 ± 0.90
Lu et al.[12] (2015)		81.80 ± 2.30
Lezama et al. [10][2] (2016)		80.69 ± 1.02
Juefei-Xu et al.[7] (2015)		78.46 ± 1.67
Jin et al. [6] (2015)		75.70 ± 2.5
VGG-Face		67.47 ± 1.73
Training and testing setups		rank-1 accuracy
k_{train}	k_{test}	
0 (base)	0 (base)	76.28 ± 2.08
72 (MMI)	0 (base)	81.46 ± 1.22
324 (synth.)	0 (base)	82.35 ± 1.59
324 (synth.)	5 (MMI)	82.64 ± 1.45
72 (MMI)	**5 (MMI)**	84.43 ± 1.26

Table 3: Results on CASIA NIR-VIS 2.0 benchmark. Lezama et al. [10][2] is in reference to the reported results of learning a low-rank embedding to the *VGG-Face* model.

Figure 7 shows the top 4 synthesis types selected for testing. We record results in Table 3 and compare with multiple state-of-the-art results on the CASIA NIR-VIS 2.0 benchmark. Notably, we are able to adapt the *VGG-Face* and drastically improve rank-1 scores from $\mathbf{67.47}\%$ to $\mathbf{84.43}\%$ and outperforming many other works cited.

5. Conclusion

We proposed approaches for intelligent synthesis selection during training and testing. These approaches exploit face synthesis methods, allowing for more efficient training and improved testing performance in constrained settings. We outlined scenarios that required a state-of-the-art DNN to be calibrated, and showed the impact of our approaches both during training and at testing. In these scenarios the trade-off between synthesizing and training with non-informative face images vs. performance is apparent, and we demonstrate the value of our optimal synthesis selection. Future work is needed to verify contributions of specific synthesis methods for different scenarios.

Acknowledgements Work was supported by NSF, NGA, ARO, and ONR.

References

[1] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. *International Conference on Machine Learning*, pages 209–216, 2007. 5

[2] M. Di Martino, G. Facciolo, and E. Meinhardt-Llopis. Poisson image editing. *Image Processing On Line*, 2016. 3

[3] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2016. 2

[4] G. Hu, X. Peng, Y. Yang, T. Hospedales, and J. Verbeek. Frankenstein: Learning deep face representations using small data. *arXiv:1603.06470*, 2016. 2, 3, 4

[5] G. 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, October 2007. 6

[6] Y. Jin, J. Lu, and Q. Ruan. Large margin coupled feature learning for cross-modal face recognition. In *International Conference on Biometrics*, pages 286–292, 2015. 8

[7] F. Juefei-Xu, D. K. Pal, and M. Savvides. NIR-VIS heterogeneous face recognition via cross-spectral joint dictionary learning and reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 141–150, 2015. 8

[8] D. E. King. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 2

[9] A. Krause, A. Singh, and G. Carlos. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008. 5

[10] J. Lezama, Q. Qiu, and G. Sapiro. Not afraid of the dark: NIR-VIS face recognition via cross-spectral hallucination and low-rank embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017. 8

[11] S. Li, D. Yi, Z. Lei, and S. Liao. The CASIA NIR-VIS 2.0 face database. In *IEEE Workshop on Perception Beyond the Visible Spectrum*, 2013. 6, 8

[12] J. Lu, V. E. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2041–2056, 2015. 8

[13] I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *IEEE European Conference on Computer Vision*, pages 579–596, 2016. 2, 4

[14] K. A. F. Mora and J. M. Odobez. Gaze estimation from multimodal Kinect data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–30, 2012. 3

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 1, 6

[16] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. Transformation pursuit for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014. 2

[17] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 3

[18] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318, 2003. 3

[19] Q. Qiu, Z. Jiang, and R. Chellappa. Sparese dictionary-based representation and recognition of human action attributes. In *IEEE International Conference on Computer Vision*, 2011. 5

[20] Q. Qiu and G. Sapiro. Learning transformations for clustering and classification. *Journal of Machine Learning Research*, 16:187–225, 2015. 5

[21] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Neural Information Processing Systems*, 2016. 2

[22] S. Saxena and J. Verbeek. Heterogeneous face recognition with cnns. In *IEEE European Conference on Computer Vision TASK-CV 2016 Workshops*, 2016. 8

[23] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 6

[24] A. Shrivastava, T. Pfister, O. Tuze, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv:1612.07828*, 2016. 2

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scal training for face identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2754, 2015. 1

[26] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 6

[27] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011. 6

[28] D. Yi, Z. Lei, and S. Li. Shared representation learning for heterogeneous face recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2015. 8