Fastfood Dictionary Learning for Periocular-Based Full Face Hallucination

Felix Juefei-Xu and Marios Savvides
CyLab Biometrics Center, Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213, USA
felixu@cmu.edu, msavvid@ri.cmu.edu

Abstract

The kernel trick becomes a burden for some machine learning tasks such as dictionary learning, where a huge amount of training samples are needed, making the kernel matrix gigantic and infeasible to store or process. In this work, we propose to alleviate this problem and achieve Gaussian RBF kernel expansion ***explicitly*** *for dictionary learning using Fastfood transform, which is an approximation of full kernel expansion. We have shown, in the context of missing data recovery through joint dictionary learning i.e. periocular-based full face hallucination, that the approximated kernel expansion using Fastfood transform for joint dictionary learning yields much better results than its image space counterparts. Also, explicit kernel expansion through Fastfood allows us to de-kernelize the reconstructed image in the feature space back to the image space, enabling applications that require reconstructive dictionaries such as cross-domain reconstruction, image super-resolution, missing data recovery, etc.*

1. Introduction

Kernel methods have seen many successes in the past, such as in kernel PCA [36], kernel Fisher discriminant analysis [38], and kernel SVM [37]. The common practice is that algorithms are reformulated to have explicit inner product form so that the kernel trick can be applied, avoiding the need of using the exact mapping explicitly. Inner products are then replaced by the kernel matrix of size $N \times N$, where N is the number of training samples. However, such strategy will become infeasible for some machine learning tasks such as dictionary learning [29], where we are trying to learn an overcomplete dictionary ($d \times M$) from the training data ($d \times N$, d is data dimension) such that they can be sparsely represented by this particular dictionary.

In this regard, the kernel **trick** has become a **burden** because the learned dictionary has far more atoms than the data dimension $M \gg d$, which means it needs way more training data to learn those atoms from, leading towards $N \gg d$, making the $N \times N$ kernel matrix gigantic and next to impossible to store or process in some applications *e.g.* a dictionary learning attempt involving over one million training samples.

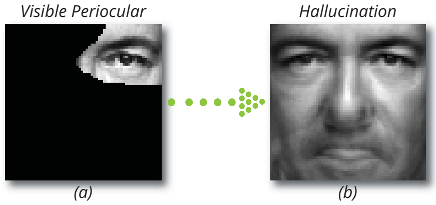


Figure 1: Quiz: can you identify this subject based on the visible periocular region (a)? How about from the hallucinated full face (b)? The answer is shown in Figure 2.

In this work, we propose to achieve Gaussian RBF kernel expansion explicitly for dictionary learning, using Fastfood transform [28], which is an approximation of full kernel expansion. Fastfood transform significantly speeds up the traditional Random Kitchen Sinks method [33] and reduces the model complexity.

We have shown, in the context of missing data recovery through joint dictionary learning *i.e.* periocular-based full face hallucination, that the approximated kernel expansion using Fastfood transform for joint dictionary learning yields much better results than its image space counterparts. Also, explicit kernel expansion through Fastfood allows us to de-kernelize the reconstructed image in the feature space back to the image space, enabling applications that require reconstructive dictionaries such as cross-domain reconstruction, image super-resolution, missing data recovery, *etc*.

Related work: Kernel dictionary learning with K-SVD [29] has been proposed to incorporate kernel methods within the paradigm of dictionary learning. The authors reformulate the entire dictionary learning problem in term of kernels, by allowing the dictionary to be a multiplication of two parts. The first one is called the "base dictionary" which houses all the training samples in feature space, and the second part is called the "coefficient dictionary" which is actually updated during dictionary learning. This kernel K-SVD method suffers from the same aforementioned problem of having to deal with the kernel matrix which is of size $N \times N$, where N is the number of training samples which can be extremely large.

One of the key challenges or disadvantages of kernel methods using full kernel expansion via kernel trick is that, no exact inverse mapping can be achieved from (high dimensional, or usually infinite dimensional) feature space back to the original image space. In the context of dictionary learning, once the kernel dictionary is learned, we cannot observe how each dictionary atoms look like because the kernel trick prohibits explicit kernel mapping and only works with inner products, leading towards the kernel matrix. Therefore, the method of [29] can only accommodate discriminative dictionaries for classification purposes, rather than reconstructive ones for tasks such as reconstructing an entire image from small amount of visible pixels, image hallucination or super-resolution, *etc*. because the learned kernel dictionary atoms have to be mapped down to the original image domain for these tasks.

Specifically, the dictionary in the feature space is decomposed as the following [29]:

$$\Phi(\mathbf{D}) = \Phi(\mathbf{X})\mathbf{A} \tag{1}$$

where $\Phi(\mathbf{X})$ is the "base dictionary" part which contains all the mapped training sample, and does not change during update. $\mathbf{A}$ is the dictionary part that is updated during learning, called the "coefficient dictionary". The final dictionary to be learned is $\mathbf{D}$. Φ is a non-linear mapping function from $\mathbb{R}^N$ to a higher dimensional feature space $\mathcal{F}$, $\Phi : \mathbb{R}^N \mapsto \mathcal{F}$. The kernel K-SVD dictionary learning is formulated as:

$$\arg\min_{\mathbf{A},\mathbf{\Gamma}} \left\| \Phi(\mathbf{X}) - \underbrace{\Phi(\mathbf{X})\mathbf{A}}_{\Phi(\mathbf{D})}\mathbf{\Gamma} \right\|_F^2 \quad \text{s.t. } \|\gamma_i\|_0 \leq K, \forall i. \tag{2}$$

Similar to the linear version of the K-SVD method [1], the optimization is carried out in an iterative fashion by first performing sparse coding with a fixed dictionary $\mathbf{A}$, and followed by dictionary updating based on the computed sparse representation $\mathbf{\Gamma}$, until convergence. The kernel version of the sparse coding is as follows:

$$\arg\min_{\gamma} \|\Phi(\mathbf{z}) - \Phi(\mathbf{X})\mathbf{A}\gamma\|_2^2 \quad \text{subject to} \quad \|\gamma\|_0 \leq K \tag{3}$$

where $\mathbf{z}$ is the input sample. The sparse coding algorithm used here is the orthogonal matching pursuit (OMP) [40] and its kernel version [29].

Contributions: The main contributions of this work include (1) achieving explicit kernel expansion for dictionary learning tasks; (2) making large-scale kernel dictionary learning feasible; (3) conducting de-kernelization that maps reconstruction in the kernel feature space back to the image space; (4) enabling applications that require reconstructive dictionaries, as opposed to discriminative ones; (5) pushing the state-of-the-art algorithm further in terms of periocular-based full face hallucination.

2. Background: Approximating Kernel

Kernel methods are used to implicitly transform an input feature space $\mathcal{Y}$ to a higher-dimensional Hilbert space $\mathcal{F}$ using the mapping function $\Phi : \mathcal{Y} \mapsto \mathcal{F}$. The kernel function k for the inner product of transformed vectors $\Phi(\mathbf{y})$ and $\Phi(\mathbf{y}')$ is defined as $k(\mathbf{y}, \mathbf{y}') = \langle \Phi(\mathbf{y}), \Phi(\mathbf{y}')\rangle$ and is often computationally expensive to evaluate for all pairs of data points. Various approximation methods have been proposed to tackle this issue. Two such approximation methods are Random Kitchen Sinks [33] and Fastfood [28]. Both methods construct k^* by explicitly mapping data points $\mathbf{y} \in \mathcal{Y}$ to a finite-dimensional space $\mathcal{Z}$ using a randomized mapping function $Z : \mathcal{Y} \mapsto \mathcal{Z}$. Z is designed such that $k^*(\mathbf{y}, \mathbf{y}') = \langle z(\mathbf{y}), z(\mathbf{y}')\rangle \approx \langle \Phi(\mathbf{y}), \Phi(\mathbf{y}')\rangle = k(\mathbf{y}, \mathbf{y}')$. In this work, we restrict our attention to approximating shift-invariant kernels, such as the Gaussian RBF kernel.

2.1. Random Kitchen Sinks

A method for generating transformed feature vectors $z(\mathbf{y}), z(\mathbf{y}') \in \mathbb{R}^{d'}$ for input vectors $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^d$ is described in [32, 33][1] such that k^* provides an unbiased estimate for k. We are dealing with shift-invariant kernels, therefore Random Kitchen Sinks can be written as a function of $\mathbf{y}-\mathbf{y}'$ and have the following property:

$$k(\mathbf{y}, \mathbf{y}') = k(\mathbf{y} - \mathbf{y}', 0) \tag{4}$$

Let $\widetilde{k}(\mathbf{y} - \mathbf{y}') = k(\mathbf{y} - \mathbf{y}', 0)$, we can write the inverse Fourier transform of $\widetilde{k}$ as a weighted sum of complex sinusoids:

$$\widetilde{k}(\mathbf{y} - \mathbf{y}') = \int_{\mathbb{R}^d} p(\omega) e^{\mathbf{i}\langle \omega, \mathbf{y}-\mathbf{y}'\rangle} d\omega \tag{5}$$

$$= \mathbb{E}_\omega \left[e^{\mathbf{i}\langle \omega, \mathbf{y}\rangle} e^{-\mathbf{i}\langle \omega, \mathbf{y}'\rangle} \right] \tag{6}$$

where $p(\omega)$ is the Fourier transform of $\widetilde{k}$ and is a proper probability distribution [32]. The authors have shown that by choosing random Fourier basis features of the form $z_\omega(\mathbf{y}) = \text{Re}\{e^{\mathbf{i}\langle \omega, \mathbf{y}\rangle}\} = \cos(\langle \omega, \mathbf{y}\rangle)$ for $\omega \sim p(\omega)$ leads to $\mathbb{E}_\omega[z_\omega(\mathbf{y}) \cdot z_\omega(\mathbf{y}')] = \widetilde{k}(\mathbf{y} - \mathbf{y}')$, and therefore k^* is an unbiased estimate for k. Since both the probability distribution $p(\omega)$ and the kernel $k(\Delta)$ are real, the integral (5) converges when the complex exponentials are replaced with cosines[2].

[1] The method was first introduced in [32] and generalized further in [33]. We refer to it with the title of the latter work [33] instead of the ambiguous phrase "random features" as in [32] for the sake of clarity.

[2] The proof is beyond the scope of this work, readers can refer to [32] and materials on harmonic analysis [34].

To generate a feature map $z(\mathbf{y})$ for $\mathbf{y}$, we simply draw d' random vectors ω from $p(\omega)$, compute each individual transformation $z_\omega(\mathbf{y})$ and concatenate the results.

The Gaussian RBF kernel defined as $\widetilde{k}(\Delta) = \exp\{-\frac{1}{2\sigma^2}\|\Delta\|_2^2\}$ has the following Fourier transform: $p(\omega) \sim \mathcal{N}(0, \frac{1}{\sigma^2})$. Computing the feature map for this kernel is straightforward:

$$z(\mathbf{y}) = [z_1(\mathbf{y}), \dots, z_{d'}(\mathbf{y})]^\top \tag{7}$$

where $z_j(\mathbf{y}) = \text{Re}\left\{\frac{1}{\sqrt{d'}}\exp\{\mathbf{i}[\mathbf{Z}\mathbf{y}]_j\}\right\}$, and $\mathbf{Z} \in \mathbb{R}^{d' \times d}$ is a random matrix whose entries are drawn i.i.d from $\mathcal{N}(0, \sigma^{-2})$.

2.2. Fastfood Feature Transform

The Fastfood feature transform [28] approximates computation of feature functions in order to compute kernel expansion in loglinear time and linear space. Typical methods to compute feature functions include Random Kitchen Sinks [33] for approximating Gaussian RBF kernel feature maps. The Fastfood approximation can also be generalized to other kernels.

The key insight is that Hadamard matrices, when combined with Gaussian scaling matrices, behave very similar to Gaussian random matrices, such as those used in the Random Kitchen Sinks method. Let $d = 2^v$ for some $v \in \mathbb{N}$ and that $d' = d$. Again, d is the number of dimensions of the input vector and d' is the number of basis functions (the dimension after kernel expansion). Fastfood replaces the random matrix $\mathbf{Z}$ in Random Kitchen Sinks with a matrix $\mathbf{V}$ defined as follows:

$$\mathbf{V} = \frac{1}{\sigma\sqrt{d}}\mathbf{SHG\Pi HB} \approx \mathbf{Z} \tag{8}$$

where $\mathbf{H} = \mathbf{H}_d$ is the $d \times d$ Walsh-Hadamard matrix, $\mathbf{\Pi} \in \{0,1\}^{d\times d}$ is a random permutation matrix, and $\mathbf{S}$, $\mathbf{G}$, and $\mathbf{B}$ are all diagonal random matrices. Specifically, $\mathbf{B}$ has uniformly drawn ± 1 values along its main diagonal, $\mathbf{G}$ has values drawn from a Gaussian distribution $\mathbf{G}_{ii} \sim \mathcal{N}(0,1)$ along its main diagonal, and $\mathbf{S}$ is a random scaling matrix.

The Fast Walsh-Hadamard transform allows the product $\mathbf{H}_d\mathbf{y}$ to be carried out in $\mathcal{O}(d\log d)$ time. This enables efficient computation of the matrix product of $\mathbf{V}$ without fully generating $\mathbf{H}$ when computing the Fastfood approximated Gaussian RBF kernel feature map.

When $d' > d$, we stack d'/d independent random matrices $\mathbf{V}_i$ via $\mathbf{V}^\top = [\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_{d'/d}]^\top$ until sufficient dimension is met and replicate (8). The feature map for Fastfood is then defined as follows:

$$\hat{\Phi}_j(\mathbf{x}) = (d')^{-\frac{1}{2}}\exp(\mathbf{i}[\mathbf{V}\mathbf{y}]_j) \tag{9}$$

Since the kernel values are real numbers, a common practice is to consider a real version of the complex feature map Φ for simplicity [28]. Thus we can replace $\Phi(\mathbf{y}) \in \mathbb{C}^{d'}$ with $\Phi'(\mathbf{y}) \in \mathbb{R}^{2d'}$ where

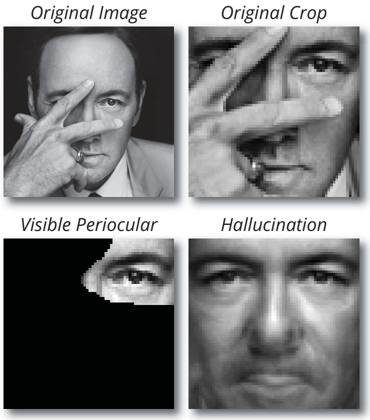


Figure 2: Periocular-based full face hallucination using the proposed Fastfood dictionary learning approach on Kevin Spacey's occluded face.

$$\Phi'_{2j-1}(\mathbf{y}) = \text{Re}\{(d')^{-\frac{1}{2}}\exp(\mathbf{i}[\mathbf{V}\mathbf{y}]_j)\} \tag{10}$$

$$= (d')^{-1/2}\cos([\mathbf{V}\mathbf{y}]_j) \tag{11}$$

$$\Phi'_{2j}(\mathbf{y}) = \text{Im}\{(d')^{-\frac{1}{2}}\exp(\mathbf{i}[\mathbf{V}\mathbf{y}]_j)\} \tag{12}$$

$$= (d')^{-1/2}\sin([\mathbf{V}\mathbf{y}]_j). \tag{13}$$

Both cosine and sine are valid choices for approximating kernel expansion with provable approximation guarantees and concentration properties [28]. In the rest of this work, we will use $\sin(\cdot)$ because it leads to a favorable property to be discussed later.

3. Proposed Method

3.1. Fastfood Dictionary Learning

Joint dictionary learning, such as through K-SVD [1] approach, has been successful in modeling non-linearity between two domains such as cross-spectral image reconstruction, image super-resolution, *etc*. In this work, we are interested in a quite novel application: reconstructing the full face from only a small number of visible pixels in the periocular region. The periocular region is the most salient facial region for face recognition purposes, and contains the richest soft-biometric cues (Figure 1) for determining age, gender, ethnicity, *etc*. [26, 25, 7, 10, 35, 23, 9, 17, 14, 8, 13, 6, 5]. Figure 2 shows a qualitative example of the proposed method, hallucinating Kevin Spacey's full face from his partially observed periocular region (17.4% visible).

Our goal is to reconstruct or hallucinate the full face given only the periocular region or even partial periocular region. We formulate this problem under a joint dictionary learning framework. We will learn a coupled dictionary where there is a sub-dictionary for the periocular region, and another for the full face, and the sparse coefficient is shared between them, leading to cross-domain reconstruction, which, in this case, is to reconstruct the full face from the periocular region. The optimization simply follows:

$$\underset{\mathbf{D},\mathbf{D}_\Lambda,\mathbf{X}}{\arg\min} \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \beta\|\mathbf{Y}_\Lambda - \mathbf{D}_\Lambda\mathbf{X}\|_F^2 \qquad (14)$$
$$\text{subject to } \forall i, \|\mathbf{x}_i\|_0 < K$$

where Λ subscript is a set indicating the dimension indices for the periocular region (subset of full face dimensions), for both the training samples and the dictionary. Here β provides a trade-off between the reconstruction error of the periocular region versus the entire face. Obtaining a consistent K-sparse encoding between the two sets of dimensions allows for a more meaningful reconstruction. Given a novel periocular image, we would first obtain the sparse representation $\mathbf{x}$ in $\mathbf{D}_\Lambda$. We then obtain the reconstruction using $\mathbf{D}\mathbf{x}$. By forcing consistent sparse representations $\mathbf{x}$ during training, we optimize for a low reconstruction error for both regions jointly and simultaneously.

Solving the formulation is achieved by a simple rearrangement before using the standard K-SVD as previously observed [3]:

$$\underset{\mathbf{D},\mathbf{D}_\Lambda,\mathbf{X}}{\arg\min} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\beta}\mathbf{Y}_\Lambda \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\beta}\mathbf{D}_\Lambda \end{pmatrix} \mathbf{X} \right\|_F^2 \qquad (15)$$
$$\text{subject to } \forall i, \|\mathbf{x}_i\|_0 \le K$$

which translates to the standard K-SVD problem:

$$\underset{\mathbf{D}',\mathbf{X}'}{\text{minimize}} \|\mathbf{Y}'-\mathbf{D}'\mathbf{X}\|_F^2 \text{ subject to } \forall i, \|\mathbf{x}_i\|_0 \le K$$

where $\mathbf{Y}' = (\mathbf{Y}^T, \sqrt{\beta}\mathbf{Y}_\Lambda^T)^T$ and $\mathbf{D}' = (\mathbf{D}^T, \sqrt{\beta}\mathbf{D}_\Lambda^T)^T$. In effect the formulation is equivalent to re-weighting dimensions belonging to Λ by $(1+\sqrt{\beta})$. We call this method dimensionally weighted K-SVD (DW-KSVD).

This cross-domain mapping is non-linear. It is worth noted that the dictionary learning process itself is non-linear due to the orthogonal matching pursuit (OMP) [40] step in the sparse coding stage, even though the image can be linearly represented by the dictionary atoms. After dictionaries for both domains are jointly learned, mapping from one domain to the other is done through sparse coding which is again non-linear. A distinction should be made between the linearity in the representation of a dictionary, and the non-linearity in dictionary learning process and the cross-domain mapping in this case. The OMP algorithm is described in Algorithm 1.

An additional degree of non-linearity can be incorporated in the Fastfood dictionary learning by transforming the training samples into features of finitely high dimension which is done by approximating the kernel expansion using a mapping function [28].

Algorithm 1 Orthogonal Matching Pursuit

```
Input: $\mathbf{y}$, $\mathbf{D}$, $\mathbf{D}^\top$, $K$
Output: $\mathbf{x}$
Initialize $\mathbf{r} \leftarrow \mathbf{y}$, $S \leftarrow \emptyset$, $i \leftarrow 0$.
repeat
    $i \leftarrow i+1$;
    Form signal proxy: $\mathbf{a} \leftarrow \mathbf{D}^\top(\mathbf{r})$;
    Find maximum element:
    $j \leftarrow \arg\max\{|a_1|, |a_2|, ..., |a_N|\}$;
    Add it to the support: $S \leftarrow S \cup \{j\}$;
    Compute least squares over support:
    $\hat{\mathbf{x}}_S \leftarrow \arg\min_\beta \|\mathbf{y} - \mathbf{D}_S\beta\|$;
    From new residue: $\mathbf{r} \leftarrow \mathbf{y} - \mathbf{D}\hat{\mathbf{x}}$;
until $i = K$ (the clipping after picking out $K$ atoms is a non-linear operation.)
```

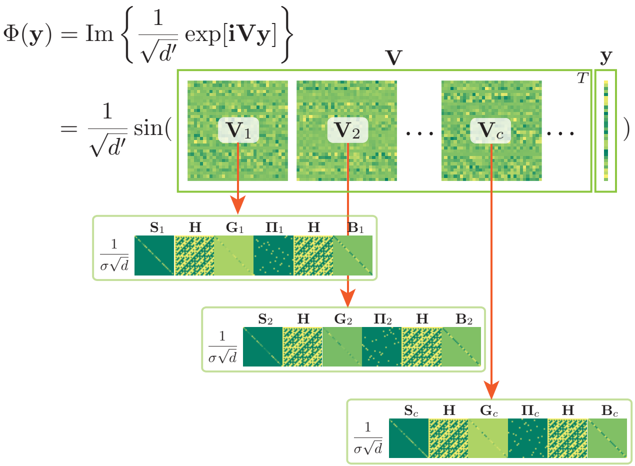

$\Phi(\mathbf{y}) = \text{Im}\left\{\frac{1}{\sqrt{d'}}\exp[\mathbf{i}\mathbf{V}\mathbf{y}]\right\}$
$= \frac{1}{\sqrt{d'}}\sin($
V
y
T
$\mathbf{V}_1$
$\mathbf{V}_2$
$\mathbf{V}_c$
. . .
. . .
)
$\frac{1}{\sigma\sqrt{d}}$
$\mathbf{S}_1$ H $\mathbf{G}_1$ $\mathbf{\Pi}_1$ H $\mathbf{B}_1$
$\frac{1}{\sigma\sqrt{d}}$
$\mathbf{S}_2$ H $\mathbf{G}_2$ $\mathbf{\Pi}_2$ H $\mathbf{B}_2$
$\frac{1}{\sigma\sqrt{d}}$
$\mathbf{S}_c$ H $\mathbf{G}_c$ $\mathbf{\Pi}_c$ H $\mathbf{B}_c$

Figure 3: Visualization of Fastfood kernel expansion $\mathbf{\Phi}(\mathbf{y})$.

The Fastfood kernel feature mapping of a training sample $\mathbf{y}$ is shown as:

$$\Phi(\mathbf{y}) = \text{Im}\left\{\frac{1}{\sqrt{d'}}\exp\{\mathbf{i}\mathbf{V}\mathbf{y}\}\right\} = \frac{1}{\sqrt{d'}}\sin(\mathbf{V}\mathbf{y}) \qquad (16)$$

where $\mathbf{V}$ is defined as $\mathbf{V} = [\mathbf{V}_1^\top, \dots, \mathbf{V}_{d'/d}^\top]^\top$ and $\mathbf{V}_c = \frac{1}{\sigma\sqrt{d}}\mathbf{S}_c\mathbf{H}\mathbf{G}_c\mathbf{\Pi}_c\mathbf{H}\mathbf{B}_c, c = 1, \dots, d'/d$. Here d' is the number of kernel basis and therefore is the dimension after kernel expansion and d is the dimension of the input training sample. $\mathbf{V}$ is constructed by concatenating blocks $\mathbf{V}_1, \dots, \mathbf{V}_{d'/d}$. Pictorial depiction is shown in Figure 3.

In the Fastfood procedure, the feature map is parametrized by the following set of vectors which are the diagonal elements of the diagonal design matrices $\mathbf{S}$, $\mathbf{G}$ and $\mathbf{B}$. These vectors are then used to construct each block $\mathbf{V}_c \in \mathbb{R}^{d \times d}$ of $\mathbf{V} \in \mathbb{R}^{d' \times d}$ as follows: for $c = 1, \dots, d'/d$,

$$s = [(s_1)^\top, \dots, (s_{d'/d})^\top]^\top, s_c = \text{diag}\{\mathbf{S}_c\} \in \mathbb{R}^d$$
$$g = [(g_1)^\top, \dots, (g_{d'/d})^\top]^\top, g_c = \text{diag}\{\mathbf{G}_c\} \in \mathbb{R}^d$$
$$b = [(b_1)^\top, \dots, (b_{d'/d})^\top]^\top, b_c = \text{diag}\{\mathbf{B}_c\} \in \{\mathbb{1}, -\mathbb{1}\}^d$$

Algorithm 2 K-SVD Dictionary Learning with Shrinkage

```
Input: $\mathbf{y}$, $\mathbf{D}$
Output: $\mathbf{D}$, $\mathbf{X}$
Task: Find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^N$ as sparse compositions, by solving $\text{minimize}_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ subject to } \|\mathbf{x}_i\|_0 \leq K, \forall i$
Initialization: Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}$ with $\ell_2$ normalized columns, set $J = 1$
repeat
  Sparse Coding Stage: Use any pursuit algorithm (such as OMP) to compute the representation vector $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of $i = 1, 2, ..., N$, $\text{minimize}_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 \text{ subject to } \|\mathbf{x}_i\|_0 \leq T_0, \forall i$
  Codebook Updating Stage:
  for each column $k = 1, 2, ..., K$ in $\mathbf{D}^{(J-1)}$ do
    Define the group of examples that use this atom, $\omega_k = \{i | 1 \leq i \leq K, \mathbf{x}_T^k(i) \neq 0\}$
    Compute the overall representation error matrix $\mathbf{E}_k$ by $\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j$
    Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to $\omega_k$ and obtain $\mathbf{E}_k^R$
    Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^\top$. Choose the updated dictionary column $\widetilde{\mathbf{d}}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1, 1)$
    $\widetilde{\mathbf{d}}_k = \text{Shrinkage}(\widetilde{\mathbf{d}}_k, 1)$.
  end for
  $J \leftarrow J + 1$
until Stopping conditions are reached
```

Now, the Fastfood DW-KSVD can be re-formulated as the following, allowing for an explicit kernel expansion of the training samples, so that the dictionary learning is carried out in the feature space. One big advantage of Fastfood expansion is that it avoids the need to operate on the kernel matrix, which for dictionary learning problems, can be gigantic and prohibitive.

$$\underset{\mathbf{D}^\Phi, \mathbf{D}_\Lambda^\Phi, \mathbf{X}}{\arg\min} \left\| \begin{pmatrix} \Phi(\zeta(\mathbf{Y})) \\ \sqrt{\beta}\Phi(\zeta_\Lambda(\mathbf{Y}_\Lambda)) \end{pmatrix} - \begin{pmatrix} \mathbf{D}^\Phi \\ \sqrt{\beta}\mathbf{D}_\Lambda^\Phi \end{pmatrix} \mathbf{X} \right\|_F^2 \quad \text{subject to} \quad \forall i, \|\mathbf{x}_i\|_0 \leq K \tag{17}$$

where $\Phi(\cdot)$ is the Fastfood kernel expansion shown in Equation 16, and ζ and ζ_Λ are zero-padding functions so that the input dimensions corresponding to the full face and the periocular can be matched to $d = 2^v$ for some $v \in \mathbb{N}$ so that Hadamard matrices can be properly constructed in Fastfood. As expected, the dimensions of the dictionary atoms in $\mathbf{D}^\Phi$ and $\mathbf{D}_\Lambda^\Phi$ conform with that of the kernel expanded input dimensions for the full face and the periocular region respectively.

3.2. De-kernelization

With the aforementioned explicit kernel expansion, the learned dictionary is going to be in feature space of dimension $d' \gg d$. For applications where a reconstructive dictionary is needed, it requires that one can have access to the learned dictionary atoms in the original image space for reconstruction purposes, as opposed to the discriminative kernel dictionary learning [29] where the learned kernel dictionary does not need to be explicitly mapped down to the original image space. The Fastfood expansion gives us an opportunity to de-kernelize the learned feature back into the image space. Recall the Fastfood transform as:

$$\Phi(\mathbf{y}) = \text{Im}\left\{ \frac{1}{\sqrt{d'}} \exp\{\mathbf{i}\mathbf{V}\mathbf{y}\} \right\} = \frac{1}{\sqrt{d'}} \overbrace{\sin(\underbrace{\mathbf{V}\mathbf{y}}_{\text{linear}})}^{\text{nonlinear}} \tag{18}$$

We now have $\Phi(\mathbf{y}) = \sin(\mathbf{V}\mathbf{y})/\sqrt{d'}$, and since $\sin(\cdot)$ is a 1-to-1 mapping within $[-\frac{\pi}{2}, \frac{\pi}{2}]$, we can re-write the Fastfood transform as:

$$\mathbf{V}\mathbf{y} = \arcsin(\sqrt{d'} \cdot \Phi(\mathbf{y})) \tag{19}$$

Since $\mathbf{V} \in \mathbb{R}^{d' \times d}$ is a tall matrix, we can apply left pseudo-inverse for obtaining the counter-part in the image space:

$$\mathbf{y} = (\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top \arcsin(\sqrt{d'} \cdot \Phi(\mathbf{y})) \tag{20}$$

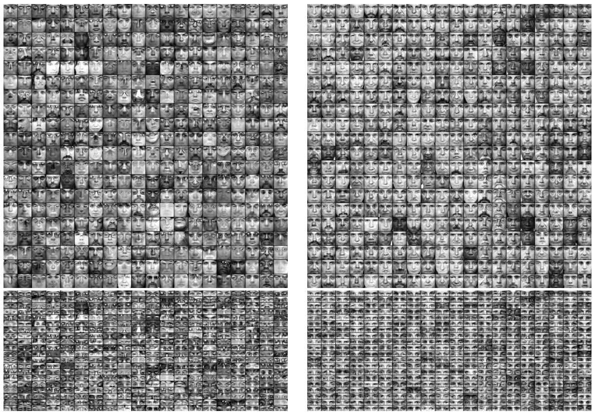
This de-kernelization procedure applies to all the representation in the feature space, be it dictionary atoms themselves, or the reconstructed image in the feature space.

After the joint dictionary is learned in feature space, we can use it for reconstructing the full face from any novel periocular input image $\mathbf{y}_\Lambda$ in the feature space. We first apply the same zero padding and Fastfood expansion to obtain its counterpart in the feature space $\Phi(\zeta_\Lambda(\mathbf{y}_\Lambda))$, and then use OMP to determine the sparse coefficient on the periocular dictionary in the feature space. The same sparse coefficient will be used to reconstruct the full face in the feature space using the full face dictionary in the feature space. Once reconstructed, all that is left is to de-kernelize the reconstruction back to the image space so that we can visualize the reconstructed full face image.

It is important that (1) the input data in the image space should be mapped to between $[-\frac{\pi}{2}, \frac{\pi}{2}]$ to ensure 1-to-1 mapping between the image space and feature space, (2) the learned dictionary atoms via K-SVD in feature space should be mapped to between $[-1, 1]$, and (3) the reconstructed image in the feature space should also be mapped to between $[-1, 1]$ so that $\arcsin(\cdot)$ wouldn't return a complex number. The modified K-SVD dictionary learning algorithm with a shrinkage operator is shown in Algorithm 2. The shrinkage operator can be defined as:

$$\text{Shrinkage}(\mathbf{d}, s) = \begin{cases} \frac{\mathbf{d} \cdot s}{|\max(y_i)|}, & \text{if } |d_i| > s, \forall i \\ \mathbf{d}, & \text{otherwise} \end{cases} \tag{21}$$

It is noteworthy that the shrinkage operation still ensures that the expected Fastfood feature map recovers the Gaussian RBF kernel, by virtue of the scaling matrix in the Fastfood expansion, see [28].



(a) (b)

Figure 4: 4(a): Full face / periocular component of the DW-KSVD dictionary. 4(b): Full face / periocular component of the de-kernelized Fastfood DW-KSVD 5x dictionary.

4. Experimental Results

Database: All test experiments were performed on the NIST's Face Recognition Grand Challenge (FRGC) ver2.0 database [31]. It has three components, the first is the generic training set which contains both controlled and uncontrolled images of 222 subjects, and a total of $12,776$ images. Second, the target set containing 466 different subjects with a total of $16,028$ images. Lastly, the probe set containing the same 466 subjects as in target set, with half as many images for each person as in the target set, bringing the total number of probe images to $8,014$.

Fastfood dictionary learning in large scale: The full face images are of size $32 \times 32 = 1024$, and the dictionary training set comprises of $N = 1,000,000$ frontal mugshot images[3]. The periocular counterpart is the top $13 \times 32 = 416$ portion of the full face images.

For Fastfood expansion, we study the 10x expansion $(d'/d = 10)$ as well as the 5x $(d'/d = 5)$ expansion of the input dimension. In addition, for the 5x case, we also include a de-kernelized version where the dictionary is immediately de-kernelized after it is learned in the feature space, and the searching of the sparse coefficient for the input periocular image is directly carried out in the image space, without using Fastfood expansion on the periocular image itself. The number of dictionary atoms to be learned is 30,000 (10x case) and 15,000 (5x case), respectively. We will denote these 3 methods as Fastfood DW-KSVD 10x, Fastfood DW-KSVD 5x, and Fastfood DW-KSVD 5x De-Kernelized (d-k). PCA, regular K-SVD, DW-KSVD in image space, are used as our benchmarks. We adopt the peak signal-to-noise ratio (PSNR) as the measurement of reconstruction fidelity between images I and I' [2, 21, 22, 18, 11, 24]. Figure 4 showcases dictionaries learned using DW-KSVD and Fastfood DW-KSVD 5x.

Table 1: Mean and standard deviations for the distributions of the PSNR values for reconstruction.

Methods	***Mean***	***Std. dev.***
PCA Recon.	12.7439	2.1288
KSVD Recon.	14.0720	2.0532
DW-KSVD Recon.	17.6402	2.3757
Fastfood DW-KSVD 5x d-k Recon.	17.7842	2.2480
Fastfood DW-KSVD 5x Recon.	18.9251	2.3213
Fastfood DW-KSVD 10x Recon.	**20.1496**	2.4243

Reconstruction fidelity: In this experiment, we reconstruct the entire target set in the FRGC ver2.0 database ($16,028$ images from 466 subjects) using the six methods (3 proposed, and 3 benchmarks) and compute the corresponding PSNR for each pair.

Figure 6 shows the overall mean PSNR computed for each subject (multiple images per subject) shown in bold line along with the mean PSNR for each individual subject shown in markers. In FRGC ver2.0 target set, each individual has on an average 34 images. Figure 7 shows the corresponding histograms. We find that methods with Fastfood dictionary learning capabilities, on average, clearly outperforms image space methods such as DW-KSVD, K-SVD and PCA by a large margin in PSNR.

Table 1 shows the mean and the standard deviation of the distribution of the PSNR values. A few randomly chosen samples and their reconstructions are shown in Figure 5. It is worth noting that most of the reconstructed faces are neutral in expression. This is because our dictionaries are trained on mugshot images, which typically have neutral expression. This, however, works in our favor because commercial matchers perform better under neutral expressions. Our proposed method actually eliminates expression variations and will be an asset for real-world matching.

Face verification: We now conduct face matching experiments using the reconstructed faces. We carry out a large-scale face verification experiment to evaluate whether the reconstructed faces can be correctly matched to their corresponding ground-truth full face under the face verification setting.

We strictly follow NIST's FRGC Experiment 1 protocol which involves 1-to-1 matching of the $16,028$ target images to themselves ($\sim$ 256 million pair-wise face match comparisons). We adopt the normalized cosine distance (NCD) to compute the similarities between images: $d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$. The NCD metric is proven to be more effective than Euclidean (ℓ_2) and Manhattan (ℓ_1) distances [30, 20, 42, 12, 39, 15, 19, 16, 41, 4]. The result of each algorithm is a similarity matrix with the size of $16,028 \times 16,028$ whose entry SimM_{ij} is the NCD between

[3] In traditional kernel dictionary learning method, one has to deal with a gigantic kernel matrix of size $1,000,000 \times 1,000,000$ which is infeasible.

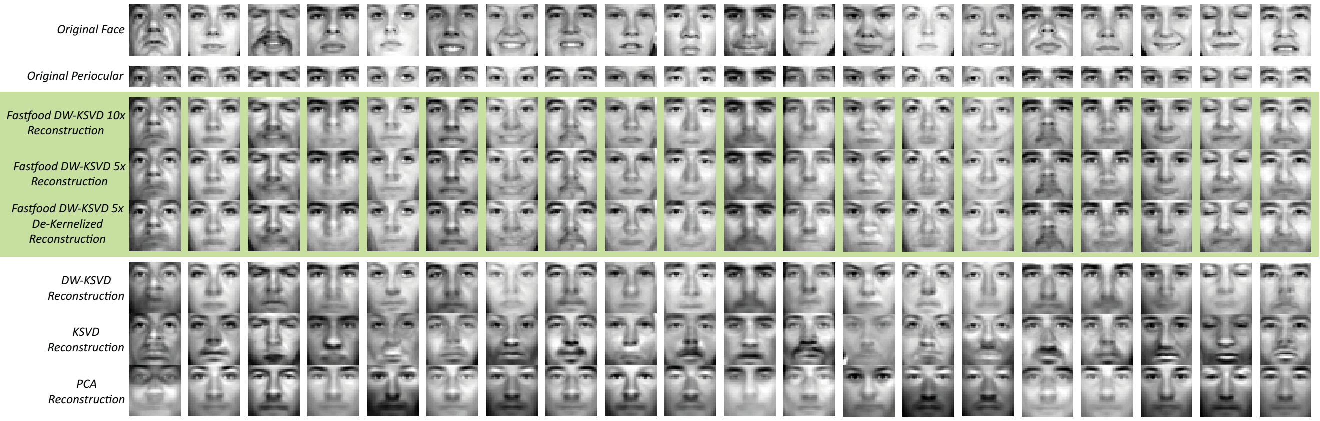

Original Face
Original Periocular
Fastfood DW-KSVD 10x Reconstruction
Fastfood DW-KSVD 5x Reconstruction
Fastfood DW-KSVD 5x De-Kernelized Reconstruction
DW-KSVD Reconstruction
KSVD Reconstruction
PCA Reconstruction

Figure 5: Original full faces and periocular region crops along with the corresponding reconstructed or hallucinated images using exclusively the periocular crops for various samples from FRGC.

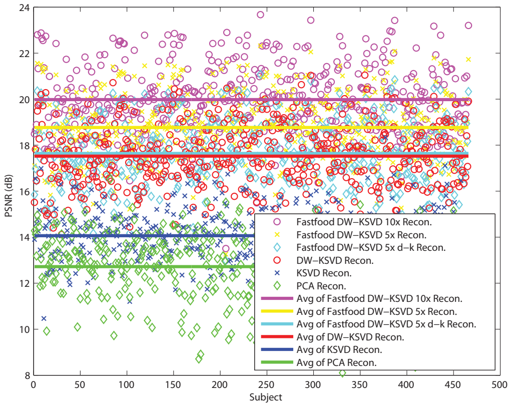


Figure 6: Mean PSNR values of reconstruction errors of individual subjects (multiple images per subject) along with the overall mean for the 6 methods.

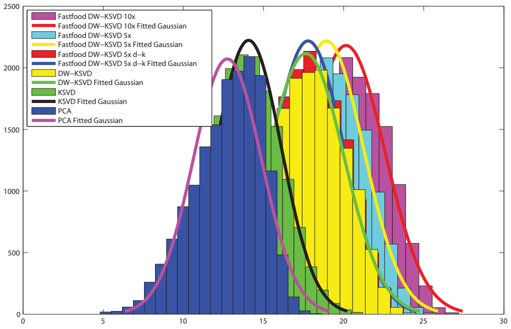


Figure 7: Overall distribution of the PSNR values for the 6 methods along with the corresponding fitted Gaussian curve.

the feature vector of query image i and gallery image j. The performance is analyzed using verification rate (VR) at 1% (0.01) false accept rate (FAR), equal error rate (EER) and the receiver operating characteristic (ROC) curves.

We use a face verification algorithm that had good performance in the NIST's FRGC evaluation: the kernel class-dependence feature analysis (KCFA) [27]. KCFA was trained on the original images of the 222 subjects belonging to FRGC ver2.0 training set. We match the original face images (gallery set) of FRGC ver2.0 target set to the corresponding reconstructed images (probe set) using the KCFA feature vectors extracted. Thus, we simulate a real-world situation, *i.e.* matching the reconstructed images to the original ones with a verification algorithm that has been trained on unseen original images.

Table 2: VR at 1% FAR and EER for the FRGC Experiment 1 evaluation using KCFA. The last 6 rows are matching reconstructed faces to the original full faces.

Methods	***VR at 1% FAR***	***EER***
Original Full Face	0.982	0.014
Fastfood DW-KSVD 10x Recon.	**0.951**	0.025
Fastfood DW-KSVD 5x Recon.	0.919	0.033
Fastfood DW-KSVD 5x d-k Recon.	0.848	0.050
DW-KSVD Recon.	0.826	0.056
KSVD Recon.	0.438	0.165
PCA Recon.	0.046	0.452

From the ROC curves in Figure 8 and the results shown in Table 2, we find that among all the methods, Fastfood dictionary learning methods performs significantly better than other image space methods.

Discussion: From both reconstruction fidelity experiments and face verification experiments, we can see that going for kernel expansion is indeed a good idea for dictionary learning. The proposed Fastfood dictionary learning approaches outperform their image space counterparts by a great margin.

Also, we notice that higher dimensional Fastfood expansion leads to better performance, which is expected. But there is always this trade-off between how high dimension the expansion should go and the performance one is achiev-

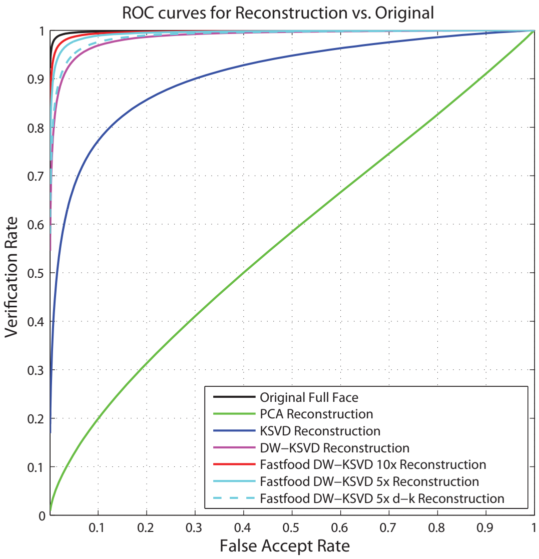


Figure 8: ROC curves obtained by matching all the reconstructed faces to the original faces using KCFA features.

ing. Remember, the whole motivation for using Fastfood expansion in kernel dictionary learning is to avoid dealing with the gigantic kernel matrix. In addition, we have also observed that it is advantageous to perform sparse coding and reconstruction in the high-dimensional feature space rather than in the original image space using de-kernelized dictionary.

5. Conclusion

In this work, we have shown how to make kernel dictionary learning feasible, and demonstrated the capability to de-kernelize the feature space representations back to the image space for reconstructive purposes. Due to the fact that for dictionary learning, the number of training samples N can be very large, and the $N \times N$ kernel matrix becomes gigantic and next to impossible to store or process. In this regard, we have proposed to achieve kernel expansion explicitly for dictionary learning, using Fastfood transform, which is an approximation of full kernel expansion. We have shown, in the context of missing data recovery through joint dictionary learning, that the approximated kernel expansion using Fastfood transform for joint dictionary learning yields much better results than its image space counterparts. Also, explicit kernel expansion through Fastfood allows us to de-kernelize the reconstructed image in the feature space back to the image space, enabling applications that requires reconstructive dictionaries such as cross-domain reconstruction, image super-resolution, missing data recovery, *etc*.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, Nov 2006. 2, 3

[2] P. Buchana, I. Cazan, M. Diaz-Granados, F. Juefei-Xu, and M.Savvides. Simultaneous Forgery Identification and Localization in Paintings Using Advanced Correlation Filters. In *IEEE ICIP*, pages 1–5, Sept 2016. 6

[3] Z. Jiang, Z. Lin, and L. Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, Nov 2013. 4

[4] F. Juefei-Xu, C. Bhagavatula, A. Jaech, U. Prasad, and M. Savvides. Gait-ID on the Move: Pace Independent Human Identification Using Cell Phone Accelerometer Dynamics. In *IEEE BTAS*, pages 8–15, Sept 2012. 6

[5] F. Juefei-Xu, M. Cha, J. L. Heyman, S. Venugopalan, R. Abiantun, and M. Savvides. Robust Local Binary Pattern Feature Sets for Periocular Biometric Identification. In *IEEE BTAS*, pages 1–8, sep 2010. 3

[6] F. Juefei-Xu, M. Cha, M. Savvides, S. Bedros, and J. Trojanova. Robust Periocular Biometric Recognition Using Multi-level Fusion of Various Local Feature Extraction Techniques. In *IEEE DSP*, pages 1–7, 2011. 3

[7] F. Juefei-Xu, K. Luu, and M. Savvides. Spartans: Single-sample Periocular-based Alignment-robust Recognition Technique Applied to Non-frontal Scenarios. *IEEE Trans. on Image Proc.*, 24(12):4780–4795, Dec 2015. 3

[8] F. Juefei-Xu, K. Luu, M. Savvides, T. Bui, and C. Suen. Investigating Age Invariant Face Recognition Based on Periocular Biometrics. In *IEEE/IAPR IJCB*, pages 1–7, Oct 2011. 3

[9] F. Juefei-Xu, D. K. Pal, and M. Savvides. Hallucinating the Full Face from the Periocular Region via Dimensionally Weighted K-SVD. In *IEEE CVPRW*, pages 1–8, June 2014. 3

[10] F. Juefei-Xu, D. K. Pal, and M. Savvides. Methods and Software for Hallucinating Facial Features by Prioritizing Reconstruction Errors, 2014. U.S. Provisional Patent Application Serial No. 61/998,043, June 17, 2014. 3

[11] F. Juefei-Xu, D. K. Pal, and M. Savvides. NIR-VIS Heterogeneous Face Recognition via Cross-Spectral Joint Dictionary Learning and Reconstruction. In *IEEE CVPRW*, pages 141–150, June 2015. 6

[12] F. Juefei-Xu, D. K. Pal, K. Singh, and M. Savvides. A Preliminary Investigation on the Sensitivity of COTS Face Recognition Systems to Forensic Analyst-style Face Processing for Occlusions. In *IEEE CVPRW*, pages 25–33, June 2015. 6

[13] F. Juefei-Xu and M. Savvides. Can Your Eyebrows Tell Me Who You Are? In *IEEE ICSPCS*, Dec 2011. 3

[14] F. Juefei-Xu and M. Savvides. Unconstrained Periocular Biometric Acquisition and Recognition Using COTS PTZ Camera for Uncooperative and Non-cooperative Subjects. In *IEEE WACV*, pages 201–208, Jan 2012. 3

[15] F. Juefei-Xu and M. Savvides. An Augmented Linear Discriminant Analysis Approach for Identifying Identical Twins with the Aid of Facial Asymmetry Features. In *IEEE CVPRW*, pages 56–63, June 2013. 6

[16] F. Juefei-Xu and M. Savvides. An Image Statistics Approach towards Efficient and Robust Refinement for Landmarks on Facial Boundary. In *IEEE BTAS*, pages 1–8, Sept 2013. 6

[17] F. Juefei-Xu and M. Savvides. Subspace Based Discrete Transform Encoded Local Binary Patterns Representations for Robust Periocular Matching on NIST's Face Recognition Grand Challenge. *IEEE Trans. on Image Proc.*, 23(8):3490–3505, aug 2014. 3

[18] F. Juefei-Xu and M. Savvides. Encoding and Decoding Local Binary Patterns for Harsh Face Illumination Normalization. In *IEEE ICIP*, pages 3220–3224, Sept 2015. 6

[19] F. Juefei-Xu and M. Savvides. Facial Ethnic Appearance Synthesis. In *Computer Vision - ECCV 2014 Workshops*, volume 8926 of *Lecture Notes in Computer Science*, pages 825–840. Springer International Publishing, 2015. 6

[20] F. Juefei-Xu and M. Savvides. Pareto-optimal Discriminant Analysis. In *IEEE ICIP*, Sept 2015. 6

[21] F. Juefei-Xu and M. Savvides. Pokerface: Partial Order Keeping and Energy Repressing Method for Extreme Face Illumination Normalization. In *IEEE BTAS*, pages 1–8, Sept 2015. 6

[22] F. Juefei-Xu and M. Savvides. Single Face Image Super-Resolution via Solo Dictionary Learning. In *IEEE ICIP*, pages 2239–2243, Sept 2015. 6

[23] F. Juefei-Xu and M. Savvides. Weight-Optimal Local Binary Patterns. In *Computer Vision - ECCV 2014 Workshops*, volume 8926 of *Lecture Notes in Computer Science*, pages 148–159. Springer International Publishing, 2015. 3

[24] F. Juefei-Xu and M. Savvides. Learning to Invert Local Binary Patterns. In *27th British Machine Vision Conference (BMVC)*, Sept 2016. 6

[25] F. Juefei-Xu and M. Savvides. Multi-class Fukunaga Koontz Discriminant Analysis for Enhanced Face Recognition. *Pattern Recognition*, 52:186–205, apr 2016. 3

[26] F. Juefei-Xu, E. Verma, P. Goel, A. Cherodian, and M. Savvides. DeepGender: Occlusion and Low Resolution Robust Facial Gender Classification via Progressively Trained Convolutional Neural Network with Attention. In *IEEE CVPRW*, June 2016. 3

[27] B. Kumar, M. Savvides, and C. Xie. Correlation pattern recognition for face recognition. *Proceedings of the IEEE*, 94(11):1963–1976, Nov 2006. 7

[28] Q. Le, T. Sarlós, and A. Smola. Fastfood - approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, Atlanta, GA, 2013. 1, 2, 3, 4, 5

[29] H. Nguyen, V. Patel, N. Nasrabadi, and R. Chellappa. Kernel dictionary learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2021–2024, March 2012. 1, 2, 5

[30] D. K. Pal, F. Juefei-Xu, and M. Savvides. Discriminative Invariant Kernel Features: A Bells-and-Whistles-Free Approach to Unsupervised Face Recognition and Pose Estimation. In *IEEE CVPR*, June 2016. 6

[31] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition. CVPR. IEEE Computer Society Conf. on*, volume 1, pages 947–954, jun 2005. 6

[32] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007. 2

[33] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 2008)*, 2008. 1, 2, 3

[34] W. Rudin. *Fourier analysis on groups*. John Wiley & Sons, 2011. 2

[35] M. Savvides and F. Juefei-Xu. Image Matching Using Subspace-Based Discrete Transform Encoded Local Binary Patterns, Sept. 2013. US Patent US 2014/0212044 A1. 3

[36] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer, 1997. 1

[37] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 1

[38] B. Scholkopft and K.-R. Mullert. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX*, 1(1):1, 1999. 1

[39] K. Seshadri, F. Juefei-Xu, D. K. Pal, and M. Savvides. Driver Cell Phone Usage Detection on Strategic Highway Research Program (SHRP2) Face View Videos. In *IEEE CVPRW*, pages 35–43, June 2015. 6

[40] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. on Information Theory*, 53:4655–4666, 2007. 2, 4

[41] S. Venugopalan, F. Juefei-Xu, B. Cowley, and M. Savvides. Electromyograph and Keystroke Dynamics for Spoof-Resistant Biometric Authentication. In *IEEE CVPRW*, pages 109–118, June 2015. 6

[42] N. Zehngut, F. Juefei-Xu, R. Bardia, D. K. Pal, C. Bhagavatula, and M. Savvides. Investigating the Feasibility of Image-Based Nose Biometrics. In *IEEE ICIP*, pages 522–526, Sept 2015. 6