Learning 2D Gabor Filters by Infinite Kernel Learning Regression

Kamaledin Ghiasi-Shirazi

Abstract
Gabor functions have wide-spread applications in image processing and computer vision. In this paper, we prove that 2D Gabor functions are translation-invariant positive-definite kernels and propose a novel formulation for the problem of image representation with Gabor functions based on infinite kernel learning regression. Using this formulation, we obtain a support vector expansion of an image based on a mixture of Gabor functions. The problem with this representation is that all Gabor functions are present at all support vector pixels. Applying LASSO to this support vector expansion, we obtain a sparse representation in which each Gabor function is positioned at a very small set of pixels. As an application, we introduce a method for learning a dataset-specific set of Gabor filters that can be used subsequently for feature extraction. Our experiments show that use of the learned Gabor filters improves the recognition accuracy of a recently introduced face recognition algorithm.

Keywords: Gabor kernels, stabilized infinite kernel learning, support vector regression

1. Introduction

Gabor functions are extensively used for feature extraction in numerous computer vision applications such as face recognition[7, 36, 17, 28, 45], image retrieval [25], palmprint recognition [34, 33], forgery detection [23], and facial expression recognition [46, 19]. Neurologists have shown that receptive field of simple cortical cells can be modeled with Gabor functions [10, 11, 29]. For this reason, computational models for visual recognition that have inspired from visual cortex use Gabor filters in the very beginning phase of feature extraction [40]. Even nowadays that deep learning methods have remarkably influenced the field of machine vision, the combination of Gabor functions and convolutional neural networks (CNN) has been observed to improve recognition rates [8].

In this paper, we show that Gabor functions are translation-invariant (also called stationary) positive-definite kernels. It is somewhat strange that this fact, despite its simple proof, has been invisible to the eyes of researchers and it is neither mentioned in classical
books such as [39, 6, 41] nor in seminal work of Genton [15] who reviewed the class of stationary kernels [1]. We believe that the positive-definiteness of Gabor functions can potentially be exploited in numerous ways for applying kernel algorithms to machine vision problems. In this paper, we target the problem of learning Gabor filters from data which in kernel methods terminology is a kernel learning problem. Perhaps the most widespread kernel learning algorithm is the multiple kernel learning (MKL) framework that seeks the best convex combination of a finite set of kernel functions [22, 8, 42, 35, 41, 21, 6]. One weakness of the MKL framework is that the initial set of kernel functions should be chosen by hand. To overcome this limitation, the infinite kernel learning (IKL) framework was introduced in which the set of initial kernels is extended to an infinite number of kernels parameterized over a continuous space [30, 11, 2, 14, 32, 31, 17]. Some of the solutions proposed for this problem were restricted to Gaussian kernels [30, 11, 2] and some were restricted to binary classification with support vector machines [14, 32, 31, 17]. However, to apply these IKL algorithms to the problem of learning Gabor functions, one should formulate the problem of learning Gabor functions as binary classification which seems to be impossible. Fortunately, Ghiasi-Shirazi [18] generalized the SIKL algorithm [17] to a more general class of machine learning problems that includes the ϵ-insensitive support vector regression (SVR). In this paper, we reduce the problem of representing an image with Gabor functions to the problem of learning a convex combination of an infinite number of Gabor kernels for regression. This gives us a mixture of Gabor functions that, when placed at positions determined by support vectors, reconstruct the given image. As a practical application of the SIKL algorithm, we propose a simple method for learning Gabor functions for a specific dataset of images from a tiny fraction of its images. However, the representation obtained by the SIKL algorithm has the problem that all Gabor functions are present at all support vector pixels. This may arouse the suspicion that the SIKL algorithm learns a universal approximator kernel function that is subsequently used by SVR for representing the input image, rejecting any link between the Gabor functions generating an image and the learned Gabor functions. In fact, we will show experimentally that the mixture of Gabor functions learned by the SIKL algorithm is approximately a highly concentrated Laplacian kernel. Using LASSO algorithm [43, we obtain a sparse representation of the original image in which the Gabor functions are located at a very sparse set of pixels. Experimental results on artificial images generated by combination of two Gabor functions confirm the potential of our sparse representation algorithm in discovering the scales, orientations, and locations of the constituting Gabor functions.

In Section 2, we give a concise and simplified introduction to SIKL regression [the general form of SIKL algorithm and its mathematical analysis can be found in 18]. We introduce our method for representing an image as a mixture of Gabor functions in Section 3. Our algorithm for choosing the parameters of Gabor filters for a specific dataset is given in Section 4. In Section 5, we show how LASSO can be utilized to obtain a sparse Gabor-based representation of an image. We experimentally evaluate the proposed method in Section 6 and conclude the paper in Section 7. In Appendix A, we give a formal proof for positive-definiteness of Gabor functions.

---

1It must be mentioned that the term "Gabor kernel" is used in computer vision literature as a synonym to "Gabor filters" and does not refer to positive-definite kernels.
2. Stabilized infinite kernel learning regression

The stabilized infinite kernel learning (SIKL) algorithm had been initially introduced in [17] for binary classification and then was generalized to more general classes of machine learning problems in [18]. In this section, we give a short introduction to SIKL regression in a simple and succinct way without going into mathematical details and without grounding the SIKL framework in its most general form. For a comprehensive introduction to the SIKL framework the reader is referred to [18].

Assume that the training set consists of the input samples \(x_1, \ldots, x_l \in \mathbb{R}^d\) and their corresponding target values \(y_1, \ldots, y_l \in \mathbb{R}\). Support vector regression attempts to learn the relation between input and output by a function of the form:

\[
f(x) = \sum_{i=1}^{l} (\hat{\beta}_i - \beta_i) k(x_i, x) + b. \tag{1}\]

The coefficients \(\hat{\beta}\) and \(\beta\) are obtained by solving the following optimization problem [9]:

\[
\begin{align*}
\max_{\hat{\beta}, \beta} & \sum_{i=1}^{l} y_i (\hat{\beta}_i - \beta_i) - \epsilon \sum_{i=1}^{l} (\hat{\beta}_i + \beta_i) \\
& - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\hat{\beta}_i - \beta_i)(\hat{\beta}_j - \beta_j) k(x_i, x_j) \\
\text{s.t.} & \quad 0 \leq \hat{\beta}_i, \beta_i \leq C \quad \text{for } i = 1, \ldots, l \\
& \quad \sum_{i=1}^{l} (\hat{\beta}_i - \beta_i) = 0
\end{align*} \tag{2}\]

where \(C \in (0, \infty]\) is a regularization constant. The above optimization problem can be rewritten in the following more succinct form:

\[
\max_{\alpha \in \mathcal{A}} - \frac{1}{2} \alpha^T A^T K A \alpha - c^T \alpha \tag{3}\]

where \(K\) is the kernel matrix obtained by applying the kernel function \(k\) to the input samples \(x_1, \ldots, x_l\),

\[
\begin{align*}
\alpha &= \begin{bmatrix} \hat{\beta} \\ \beta \end{bmatrix}, \\
A &= \begin{bmatrix} I_l & -I_l \end{bmatrix}^T, \\
c &= \begin{bmatrix} y_1 & y_2 & \cdots & y_l - y_1 & -y_2 & \cdots & -y_l \end{bmatrix}^T - \epsilon, \\
\mathcal{A} &= \left\{ \alpha | \alpha = \begin{bmatrix} \hat{\beta} \\ \beta \end{bmatrix}, 0 \leq \alpha_i \leq C, \sum_{i=1}^{l} (\hat{\beta}_i - \beta_i) = 0 \right\}.
\end{align*} \tag{4}\]
and $I_ℓ$ is the identity matrix of size $ℓ$.

Consider the set of kernels $\{k_{γ} : γ ∈ Γ\}$, where $Γ$ is a continuously parameterized index set. Let $P(Γ)$ be the set of all probability measures on $Γ$. It can be shown [see 30] that for any probability measure $p ∈ P(Γ)$, the function

$$\hat{k}_p(x, z) = \int Γ k_γ(x, z) \, dp(γ)$$

(5)

is a convex combination of the set of kernels $\{k_γ : γ ∈ Γ\}$. Conversely, any convex combination of the set of kernels $\{k_γ : γ ∈ Γ\}$ can be written in the form of Eq. (5). In the IKL framework, it is assumed that $Γ$ is a compact Hausdorff space (e.g. a bounded and closed subset of $R^2$) and the problem is to find the best kernel in the form of Eq. (5). The SIKL framework relaxes the assumption on $Γ$ to locally-compact Hausdorff spaces (e.g. $R$ or $R^2$). For mathematical concreteness and for provisioning a mechanism to control the capacity of the learning machine, the SIKL framework introduces a vanishing function $G(γ) : Γ → [0, 1]$ into the framework. The stabilized convex combination of the kernels $\{k_γ : γ ∈ Γ\}$ with stabilizer $G(γ)$ and probability measure $p ∈ P(Γ)$ is defined as:

$$\bar{k}_p(x, z) = \int Γ k_γ(x, z) G(γ) \, dp(γ).$$

(6)

Correspondingly, the set of stabilized convex combination of kernels $\{k_γ : γ ∈ Γ\}$ with stabilizer $G(γ)$ is defined as:

$$\mathcal{K} := \{\bar{k}_p : p ∈ P(Γ)\}.$$  

(7)

The problem of simultaneously learning the regression function along with a kernel function $\bar{k}_p ∈ \mathcal{K}$ can be formulated as:

$$\sup_{p ∈ P(Γ)} \min_{α ∈ A} \left\{ \int_{γ ∈ Γ} [G(γ)α^T A^T K(γ)Aα + c^T α] \, dp(γ) \right\}$$

(8)

where $K(γ)$ is the kernel matrix that is obtained by applying the kernel function $k_γ$ to the input training data $x_1, ..., x_l$.

Ghiasi-Shirazi [18] proved that the probability measure that optimizes the above problem is discrete with finite support. The SIKL toolbox optimizes the above problem by semi-infinite programming and returns the weights $μ_i$ and the parameters $γ_i ∈ Γ$ which identify the optimal kernel $\bar{k}$ by the following formula:

$$\bar{k}(x, z) = \sum_{i=1}^m μ_i k_{γ_i}(x, z).$$

(9)

---

2For metric spaces like $R$ and $R^2$ this means that the function tends to zero at infinity. For introduction to vanishing functions in general topological spaces please see [page 70 of 37].
We added the Gabor kernel to the SIKL toolbox and exploited some special properties of Gabor kernels to optimize the toolbox. Specifically, since Gabor kernels are two-dimensional, we modified the global optimization algorithm of SIKL to search the space of parameters systematically.

3. Gabor-based image representation using SIKL

In this section, we show how SIKL regression can be applied to the task of image representation by Gabor functions. We consider the following form for Gabor functions which is essentially a slightly modified version of the form chosen by [36]:

\[
\Psi(x, y; x_0, y_0, \omega, \theta) = e^{-\omega^2((x-x_0)^2+(y-y_0)^2)/8\pi^2} \cos(\omega((x-x_0)\cos \theta + (y-y_0)\sin \theta))
\]

where the point \((x_0, y_0)\) is the center of the Gabor function in the spatial domain and the parameters \(\omega\) and \(\theta\) determine the scale and orientation of Gabor function, respectively. Note that, in Eq. (10), the only inputs are \(x\) and \(y\) and \(x_0\) and \(y_0\) are parameters of the Gabor function. By considering \(x_0\) and \(y_0\) as inputs, we arrive at the following definition for Gabor kernels:

\[
k_{\omega, \theta}([x, y], [x', y']) = \Psi(x, y; x', y', \omega, \theta).
\]

There is another parameterization for Gabor functions which is obtained from Eq. (10) by setting \(\omega = \pi/2\) and \(\theta = \mu \pi/8\). This \(\mu \nu\)-parameterization is specially important since manual selection of Gabor parameters is usually done in that form. We use this form when a parameter is to be chosen by hand or when reporting the learned parameters of Gabor functions. Appendix A elaborates on the chosen form for Gabor functions and gives a proof for positive-definiteness of Gabor kernels.

Now, assume that we want to search for the best convex combination of Gabor kernels whose scale parameters are in the range \([\omega_{\ell_0}, \omega_{u_1}]\). This choice corresponds to a rectangular vanishing function in SIKL formulation which is not appropriate due to the jumps from 0 to 1 and vise versa. Therefore, we choose the following trapezoidal stabilizing function:

\[
G(\omega, \theta) = \left\{ \begin{array}{ll}
0 & \omega < \omega_{\ell_0} \\
\omega_{\ell_0} - \omega_{\ell_0} & \omega_{\ell_0} < \omega < \omega_{\ell_1} \\
\omega_{\ell_1} - \omega_{\ell_0} & \omega_{\ell_1} < \omega < \omega_{u_1} \\
\omega_{u_1} - \omega_{u_0} & \omega_{u_1} < \omega < \omega_{u_0} \\
0 & \omega > \omega_{u_0}
\end{array} \right.
\]

The stabilized convex combination of Gabor functions with stabilizer \(G(\omega, \theta)\) and probability measure \(p\) is defined as:

\[
\tilde{k}_p([x, y], [x', y']) = \int_{\mathbb{R}^2} k_{\omega, \theta}([x, y], [x', y']) G(\omega, \theta) \, dp(\omega, \theta).
\]
Consequently, the set of stabilized convex combination of Gabor functions with stabilizer \( G(\omega, \theta) \) can be expressed as:

\[
\mathcal{K} := \{ \bar{k}_p : p \in \mathcal{P}(\mathbb{R}^2) \}.
\] (14)

As stated previously, although the optimization is over a continuous space of parameters, the optimal kernel has a finite expansion of the form:

\[
\bar{k}([x, y], [x', y']) = \sum_{i=1}^{m} \mu_i k_{\omega_i, \theta_i}(x - x', y - y').
\] (15)

For a given image \( I \), we generate a training set that consists of positions of pixels as input and the intensity at those pixels as desired outputs. We then use the SIKL regression algorithm to learn the above kernel and the parameters of a SVR machine simultaneously in order to predict the intensity of each pixel correctly. The solution of the SIKL problem gives the number of participating kernels \( m \), Gabor parameters \( \omega_i \) and \( \theta_i \) for \( i = 1, \ldots, m \), and the support vector coefficients \( \hat{\beta}_i, \beta_i \) for \( i = 1, \ldots, \ell \), where \( \ell \) is the number of pixels in the image, such that:

\[
I(x, y) \approx \sum_{i \in \mathcal{SV}} (\hat{\beta}_i - \beta_i) \sum_{j=1}^{m} \mu_j k_{\omega_j, \theta_j}(x - x_i, y - y_i) + b.
\] (16)

This representation signifies the Gabor functions that are contributing to the construction of the input image \( I \).

4. Learning dataset-specific Gabor filters

When Gabor filters are used for feature extraction from a dataset, their parameters are usually tuned by hand and it is customary to use 40 Gabor functions with 5 scales and 8 directions\[^{[27, 26, 36, 20]}\]. However, since Gabor functions are defined over a pixel-space, appropriate choice of their parameters is sensitive to the resolution of the images. In Section 3, we proposed an algorithm for learning an image representation based on Gabor functions by SIKL. It is an accepted practice in machine learning that the first phases of information processing usually model the distribution of the input data while the task of discrimination is assigned to higher layers \[^{[5, 12]}\]. So, we take the assumption that the Gabor functions that are appropriate for representing an image, can also be used for feature extraction. By clustering the parameters obtained from a small fraction of images from a dataset using the k-means algorithm, we obtain a set of Gabor functions that are appropriate for representing any image in that dataset. Dataset-specific details on our method for learning Gabor filters for CMU-PIE and EYaleB datasets are given in Section 6.1.
5. Sparse image representation using Gabor kernels

The Gabor kernels learned by the method proposed in the previous section are global in the sense that each kernel is present at every location. In Section 6.2 we show that the mixture of the learned Gabor functions is approximately a concentrated Laplacian kernel. It may be questioned whether the Gabor kernels learned by the SIKL algorithm are those that are actually participating in the generation of an image or the learned combined concentrated Laplacian kernel acts as a universal approximator function that can be utilized by the SVR machine for approximating any input image. In this section, we aim to represent an image sparsely by a combination of Gabor functions such that each Gabor function is located at a small number of pixels. It has the benefit that it associates Gabor functions to the specific locations at which they are present. This problem has been previously considered by Fischer et al. [13] who proposed an algorithm based on local competition. It must be mentioned that the set of Gabor functions chosen by the SIKL algorithm is already sparse. This sparseness is the result of the implicit $L_1$ constraint $\|p\|_1 = 1$ over the probability measure $p$ in Eq. (8) which holds since the Lebesgue integral of any probability measure is 1. Thus, we assume that all the Gabor kernels that are found by the SIKL algorithm should be present in the sparse representation as well. We then try to sparsify the set of pixels at which each kernel is present. We start from Eq. (16) obtained in the previous section. By exchanging the order of summation we obtain:

$$I(x, y) \approx \sum_{j=1}^{m} \sum_{i \in SV} [\mu_j(\hat{\beta}_i - \beta_i)] k_{\omega_j, \theta_j}(x - x_i, y - y_i) + b.$$  \hspace{1cm} (17)

Our goal is to approximate the inner summation with a sparse combination of the training input data. Let $b^j$ be an $\ell \times 1$ vector whose $n$’th element is:

$$b^j_n = \sum_{i \in SV} [\mu_j(\hat{\beta}_i - \beta_i)] k_{\omega_j, \theta_j}(x_n - x_i, y_n - y_i).$$  \hspace{1cm} (18)

Assume $K^j$ is the $\ell \times \#sv$ kernel matrix associated with the kernel function $k_{\omega_j, \theta_j}$ in which rows correspond to the image coordinates and columns correspond to the support vector image coordinates. According to Eq. (17), to obtain a sparse representation for image $I$, we should find a sparse vector $\rho^j$ such that, for $n = 1, ..., \ell$, we have:

$$b^j_n = \sum_{i \in SV} [\mu_j(\hat{\beta}_i - \beta_i)] k_{\omega_j, \theta_j}(x_n - x_i, y_n - y_i) \approx \sum_{i \in SV} \rho^j_i k_{\omega_j, \theta_j}(x_n - x_i, y_n - y_i).$$  \hspace{1cm} (19)

Eq. (19) can be written in the matrix notation as:

$$b^j \approx K^j \rho^j.$$  \hspace{1cm} (20)
We have:

\[
I(x_n,y_n) \approx \sum_{j=1}^{m} \sum_{i \in SV} \left[ \mu_j (\hat{\beta}_j - \beta) \right]_k \omega_j \left( x_n - x_i, y_n - y_i \right) + b
\]

\[
= \sum_{j=1}^{m} b_n^j + b \approx \sum_{j=1}^{m} K_{n*}^j \rho^j + b
\]

\[
= \sum_{j=1}^{m} \sum_{i \in SV} \rho^j_k \omega_j \left( x_n - x_i, y_n - y_i \right) + b
\]

where \( K_{n*}^j \) is the n’th row of the kernel matrix \( K^j \) and sparseness of this representation follows from the sparseness of the vector \( \rho^j \). To find a sparse vector \( \rho^j \) that satisfies Eq. (20), we use LASSO [43] which solves the following optimization problem:

\[
\min_{\rho^j} \left\{ \frac{1}{T} \| b - K^j \rho^j \|^2_2 + \lambda \| \rho^j \|_1 \right\}.
\]

To undo the negative effect of the \( L_1 \) regularization term in LASSO on the quality of approximation of Eq. (20), we again solve Eq. (20) using the least squares method with the constraint that the pattern of sparseness of \( \rho^j \) found by LASSO would be preserved.

6. Experiments

The experiments of this section are designed with two goals in mind. First, we want to analyze the proposed algorithm in details and discover the nature of the learned Gabor functions. Second, we want to show the usefulness of the proposed method in automatic learning of Gabor functions for a given dataset. In Section 6.1, we report our experiments on the application of the learned Gabor functions to the face recognition problem and show that it yields favorable recognition accuracy over a hand-tuned choice made by experts. In Section 6.2, we analyze the learned Gabor functions and show that the weighted combination of learned Gabor kernels is equivalent to a concentrated Laplacian kernel. Finally, in Section 6.3, we analyze the proposed algorithm for Gabor-based sparse representation of images.

6.1. Selection of Gabor filter for face recognition

In this section, we want to show that use of Gabor filters learned by the method proposed in Section 4 can increase the accuracy of machine vision applications compared with Gabor filters chosen by hand. For this purpose, we chose the MOST system that is recently proposed by Ren et al. [36] for the task of face recognition and uses Gabor filters for feature extraction. The code of the MOST algorithm along with the CMU-Light and EYaleB face datasets were obtained by contacting Ren et al. [36]. CMU-Light is the name [36] gave to the illumination part of CMU-PIE dataset [4] which consists of 43 images captured at different illumination conditions from 68 persons, amounting to 2924 images. The Extended Yale B dataset [16], which is abbreviated as EYaleB, consists
of 64 frontal images from 38 persons again taken at different illumination conditions, amounting to 2432 images. Ren et al. [36] removed the 5 most dark images from the original 64 instances provided for each person in Extended Yale B dataset. In addition, all images had been histogram-equalized and were resized to width 46 and height 56.

Since the SIKL algorithm is time-consuming, considering the locality of Gabor functions, instead of representing a whole image with Gabor functions, we break images into several smaller (sometimes overlapping) regions and represent each region with a set of Gabor functions. From each face, we extract four regions around the two eyes, the nose, and the mouth (see Figure 1). We used a trapezoidal vanishing function in the formulation of the SIKL algorithm with $\omega_{\ell_0} = \frac{\pi}{512}$, $\omega_{\ell_1} = \frac{\sqrt{2}\pi}{512}$, $\omega_{u_1} = 2\sqrt{2}\pi$, and $\omega_{u_0} = 4\pi$. In $\mu\nu$-space, these choices correspond to $\nu_{\ell_0} = 16$, $\nu_{\ell_1} = 15$, $\nu_{u_1} = -5$, and $\nu_{u_0} = -6$.

From each dataset, we randomly selected 28 images (which in both cases amounts to less than 2% of the data) for learning the parameters of Gabor filters. We used the kmeans algorithm to cluster the Gabor parameters extracted from these 28 images to obtain 40 Gabor filters. Finally, we evaluated the original and learned Gabor filters on the task of face recognition using the MOST method. The number of training images used by the MOST algorithm, called $n_{\text{train}}$, is an important factor in the accuracy of the face recognition system. We compare the accuracies obtained by the original 40 filters used by Ren et al. [36] and the 40 filters learned by our method. Each experiment is repeated 30 times. The results of these experiments are summarized in Table 1. As can be seen, when the number of training images for the MOST algorithm is low, use of the learned Gabor filters significantly increases the recognition rate. Figure 2 shows the parameters of the original and the learned filters in the $\mu\nu$-space. It is clear from the figure that the parameters of Gabor filters used by Ren et al. [36] do not cover the whole region of parameters that are indeed required for representing images with Gabor functions. In addition, the dataset-specific distributions for Gabor kernel parameters depicted in Figures 2.b and 2.c can be used as a guideline for manual tuning of parameters of Gabor filter.

6.2. Analysis of the learned Gabor functions

In Section 3, we showed how the SIKL algorithm can be exploited for representing an image with Gabor kernels. The learned representation can be equivalently obtained by support vector regression with a single kernel that is the weighted combination of the selected Gabor kernels (see Eq. 15). An interesting question is what is the single kernel that is equivalent to the combination of the learned Gabor functions. We answer this question empirically by drawing the shape of the combined kernel. Figure 3 shows the combined kernels for two sample images from CMU-Light and EYaleB datasets. As can be seen, the weighted combination of the learned Gabor functions is approximately a concentrated Laplacian kernel.

6.3. Discovering locations of constituting Gabor functions

In this section, we experimentally evaluate the SIKL+LASSO algorithm of Section 5 in discovering the exact location of Gabor functions participating in a sparse representation of an image. For this purpose, we first produced a few artificial images by combination of two randomly generated Gabor functions. Figure 4.a shows several examples of these artificially generated images. Then, we used the SIKL+LASSO method
Table 1: Comparison of accuracies obtained by MOST algorithm on CMU-Light and EYaleB datasets when using the manually tuned Gabor filters of Ren et al. [36] and when using Gabor filters learned by the proposed algorithm. The parameter \( \text{ntrain} \) refers to the number of training faces used by the MOST face recognition algorithm [36]. The proposed method uses less than 2% of images of each dataset for learning the parameters of Gabor kernels. For each dataset, the last column shows the two-tailed P-values for paired t-test. Results that are statistically significant are bold-faced. It must be emphasized that since a paired t-test is used, P-values cannot be computed from statistics summarized in this table.

| ntrain | CMU-Light | EYaleB |
|--------|-----------|--------|
|        | manually tuned | learned | manually tuned | learned |
| 1      | 79.13 ± 5.7001 | 81.66 ± 5.62 | 37.65 ± 2.99 | 39.96 ± 3.89 |
| 2      | 91.15 ± 5.8534 | 92.18 ± 5.37 | 72.99 ± 2.88 | 73.71 ± 2.90 |
| 3      | 97.39 ± 4.0537 | 97.54 ± 3.74 | 88.22 ± 2.45 | 88.46 ± 2.03 |
| 4      | 98.95 ± 2.0655 | 98.90 ± 1.91 | 94.42 ± 1.72 | 94.36 ± 1.55 |
| 5      | 99.47 ± 1.1044 | 99.36 ± 1.13 | 97.09 ± 1.00 | 96.99 ± 0.96 |

proposed in Section 5 for discovering the positions of the original Gabor functions. We used a regularization constant of \( \lambda = 0.1 \) for the LASSO algorithm. Figure 4b shows the approximations of images of Figure 4a generated by the SIKL algorithm. The set of support vector pixels found by the SIKL algorithm are depicted in Figure 4c. The approximations of the images of Figure 4a generated by the SIKL+LASSO algorithm along with the positions of the discovered Gabor functions are shown in Figure 4d. As can be seen, both the SIKL algorithm of Section 3 and the SIKL+LASSO algorithm of Section 5 generate acceptable approximations to the original images. On the other hand, while the set of support vectors obtained by the SIKL algorithm contains many pixels, the SIKL+LASSO algorithm has been successful in obtaining a very sparse representation of the images. However, in some cases the Gabor functions learned by the SIKL+LASSO algorithm do not correspond exactly to the generating ones. It must be mentioned that since Gabor functions constitute an overcomplete system, it is natural that an image can be represented by different combinations of these functions. Noting that the SIKL+LASSO method uses exactly those Gabor kernels that had been obtained by the SIKL method, this experiment reveals that the set of Gabor kernels learned by the SIKL algorithm is strongly related to those generating an image.

7. Conclusion

In this paper, we exploited the fact that a practical form of Gabor functions is also a positive-definite kernel to find an image representation based on Gabor functions. This
representation is learned by the stabilized infinite kernel learning regression algorithm that had been previously proposed by Ghiasi-Shirazi [15]. The obtained representation has the weakness that the learned Gabor kernels are not localized and are present at all pixels. We proposed a sparse representation algorithm based on LASSO and showed that in simple cases it can recover the underlying generating Gabor functions of images. As an application of our method, we proposed an algorithm for automatic learning of parameters of Gabor filters in the task of face recognition. Our experiments on CMU-PIE and Extended Yale B datasets confirm the usefulness of the proposed algorithm in automatic learning of Gabor filters.

8. Acknowledgment

The author wishes to express appreciation to Research Deputy of Ferdowsi University of Mashhad for supporting this project by grant No.: 2/38449. The author thanks Chuan-Xian Ren for providing him with the code of the MOST algorithm [36] and the processed versions of CMU-PIE and Extended Yale B datasets. The author also thanks his colleagues, Ahad Harati and Ehsan Fazl-Ersi for their valuable comments.

References

[1] A. Argyriou, C.A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In Proceedings of the 18th Conference on Learning Theory, volume 18, pages 338–352, 2005.
[2] A. Argyriou, R. Hauser, C.A. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. In Proceedings of the 23rd International Conference on Machine Learning, volume 23, pages 338–352, Pittsburgh, PA, 2006.
[3] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In Proceedings of the 21st International Conference on Machine Learning, volume 21, pages 41–48, Banff, Canada, 2004. Omnipress.
[4] S Baker, T Sim, and M Bsat. The cmu pose, illumination, and expression database. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(12):1615–1618, 2003.
[5] Christopher M Bishop. Neural networks for pattern recognition. Oxford university press, 1995.
[6] Serhat S Buçak, Rong Jin, and Anil K Jain. Multiple kernel learning for visual object recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7):1354–1369, 2014.
[7] L.A. Cament, F.J. Galdames, K.W. Bowyer, and C.A. Perez. Face recognition under pose variation with local gabor features enhanced by active shape and statistical models. Pattern Recognition, 48 (11):3371–3384, 2015. doi: 10.1016/j.patcog.2015.05.017.
[8] Shuo-Yiin Chang and Nelson Morgan. Robust cnn-based speech recognition with gabor filter kernels. In INTERSPEECH, pages 905–909, 2014.
[9] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines. Cambridge University Press, 2000.
[10] John G Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. Vision Research, 20(10):847–856, 1980.
[11] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. JOSA A, 2(7):1160–1169, 1985.
[12] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11(Feb):625–660, 2010.
[13] Sylvain Fischer, Gabriel Cristóbal, and Rafael Redondo. Sparse overcomplete gabor wavelet representation based on local competitions. IEEE Transactions on Image Processing, 15(2):265–272, 2006.
[14] P. V. Gehler and S. Nowozin. Infinite kernel learning. In Proceedings of the NIPS 2008 Workshop on "Kernel Learning: Automatic Selection of Optimal Kernels", pages 1–4, 2008.
[15] M.G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

[16] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

[17] Kamaledin Ghiasi-Shirazi, Reza Safabakhsh, and Mostafa Shamsh. Learning translation invariant kernels for classification. *Journal of Machine Learning Research*, 11(Apr):1353–1390, 2010.

[18] Sayed Kamaledin Ghiasi-Shirazi. Learning kernel functions based on a stabilized convex combination model. PhD thesis, Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), 2011. (In Farsi language).

[19] W. Gu, C. Xiang, Y.V. Venkatesh, D. Huang, and H. Lin. Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *Pattern Recognition*, 45(1):80–91, 2012. doi: 10.1016/j.patcog.2011.05.006.

[20] Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb. Identification using encrypted biometrics. In *International Conference on Computer Analysis of Images and Patterns*, pages 440–448. Springer, 2013.

[21] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research*, 12(Mar):953–997, 2011.

[22] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[23] J.-C. Lee. Copy-move image forgery detection based on gabor magnitude. *Journal of Visual Communication and Image Representation*, 31:320–334, 2015. doi: 10.1016/j.jvcir.2015.07.007.

[24] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[25] S ¨Oz¨o˘g¨ur-Aky¨uz and G-W Weber. Infinite kernel learning via infinite and semi-infinite programming. *Optimisation Methods & Software*, 25(6):937–970, 2010.

[26] S ¨Oz¨o˘g¨ur-Aky¨uz and G-W Weber. On numerical optimization theory of infinite kernel learning. *Journal of Global Optimization*, 48(2):215–239, 2010.

[27] X. Pan and Q.-Q. Ruan. Palmprint recognition using gabor feature-based (2d)2pca. *Neurocomputing*, 71(13-15):3032–3036, 2008. doi: 10.1016/j.neucom.2007.12.030.

[28] X. Pan and Q.-Q. Ruan. Palmprint recognition using gabor-based local invariant features. *Neurocomputing*, 72(7-9):2040–2045, 2009. doi: 10.1016/j.neucom.2008.11.019.

[29] Frédéric Rakotomamonjy, F.R. Bach, S. Canu, and V. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[30] Chun-Xian Ren, Dao-Qing Dai, Xiao-Xin Li, and Zhao-Rong Lai. Band-rewighted kernel embedding for face image representation and recognition. *IEEE Transactions on Image Processing*, 23(2):725–740, 2014.

[31] W. Rudin. *Real & Complex Analysis, 3rd edition*. McGraw-Hill, New York, 1987.

[32] Saeed Saremi, Terrence J Sejnowski, and Tatyana O Sharpee. Double-gabor filters are independent components of small translation-invariant image patches. *Neural computation*, 25(4):922–939, 2013.

[33] B. Schölkopf and A. Smola. *Learning with Kernels- Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[34] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomas Poggio. Robust object recognition with cortex-like mechanisms. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):411–426, 2007.

[35] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
Appendix A. Gabor functions as positive definite kernels

Two dimensional Gabor functions have been studied in depth by Lee [24]. He starts from a general form of Gabor functions that consists of 8 parameters as follows:

$$\Psi(x, y; x_0, y_0, \xi_0, \nu_0, \rho, \sigma, \beta) = \frac{1}{\sqrt{\pi \sigma \beta}} e^{-\left(\left(\left(x-x_0\right)\cos\theta + \left(y-y_0\right)\sin\theta\right)^2 + \left(-\left(x-x_0\right)\sin\theta + \left(y-y_0\right)\cos\theta\right)^2\right) / \sigma^2 \beta^2} \times e^{i\left(\xi_0(x-x_0) + \nu_0(y-y_0) + \rho\right)}$$

(A.1)

where the pair \((x_0, y_0)\) is the center of the filter in spatial domain, the parameters \(\beta, \sigma, \theta\) determine an elliptical Gaussian, the parameters \(\xi_0, \nu_0\) are the horizontal and vertical frequencies, and \(\rho\) is the phase parameter. He then simplifies the form of Gabor functions by setting \(\rho = 0\). Note that, even after setting \(\rho = 0\), the above form of Gabor functions is undesirably too general and includes both Gaussian filters (when frequency parameters are zero) and sinusoidal waves (when \(\sigma, \beta \to \infty\)). Considering the biological observations reported about the biological visual cells, Lee [24] reduces the number of parameters one by one until he arrives at a form with only two parameters. Similar two parameter forms for Gabor functions have been used by other researchers [27, 26, 36, 20, 38]. All of these forms are special cases of Eq. (A.1) with \(\rho = 0\). We now prove that Gabor functions are positive definite kernels.

Proposition 1. The complex-valued Gabor function \(k([x, y], [x', y'])\) defined by the following equation is a positive definite kernel:

$$k([x, y], [x', y']) = \frac{1}{\sqrt{\pi \sigma \beta}} e^{-\left(\left(\left(x-x'\right)\cos\theta + \left(y-y'\right)\sin\theta\right)^2 + \left(-\left(x-x'\right)\sin\theta + \left(y-y'\right)\cos\theta\right)^2\right) / \sigma^2 \beta^2} \times e^{i\left(\xi_0(x-x') + \nu_0(y-y')\right)}$$

(A.2)
Proof. Since the leading coefficient \( \frac{1}{\sqrt{\pi \sigma \beta}} \) is positive and the class of positive definite kernels is closed under multiplication, it is enough to prove that the following functions are positive definite:

\[
\begin{align*}
  k_1([x,y],[x',y']) &= \exp \left( -\frac{(x-x') \cos \theta + (y-y') \sin \theta}{2\sigma^2} \right) \\
  k_2([x,y],[x',y']) &= \exp \left( -\frac{-(x-x') \sin \theta + (y-y') \cos \theta}{2\beta^2} \right) \\
  k_3([x,y],[x',y']) &= e^{i(\xi_0(x-x') + \nu_0(y-y'))}.
\end{align*}
\]

(A.3)

The functions \( k_1 \) and \( k_2 \) are positive definite since they are Gaussian functions with general covariance matrices. Considering the fact that any function of the form \( k(z,z') = f(z)f(z') \) is positive definite, positive definiteness of \( k_3 \) follows from the following equation:

\[
k_3([x,y],[x',y']) = e^{i(\xi_0(x-x') + \nu_0(y-y'))} e^{-i(\xi_0 x' + \nu_0 y')}
\]

(A.4)

Corollary 1. Any real-valued Gabor function expressible in the following form is a positive definite kernel.

\[
k([x,y],[x',y']) = \frac{1}{\sqrt{\pi \sigma \beta}} e^{-\left( \frac{(x-x') \cos \theta + (y-y') \sin \theta}{2\sigma^2} \right) \left( \frac{-(x-x') \sin \theta + (y-y') \cos \theta}{2\beta^2} \right)} \\
\times \cos(\xi_0(x-x') + \nu_0(y-y'))
\]

(A.5)

Proof. This follows from the fact that the real part of a complex-valued positive definite kernel function is a real-valued positive definite kernel [see page 31].
Figure 1: Regions for extracting subimages from face images. Note that two regions are overlapping. (a) and (b): An image from CMU-Light dataset and its associated subimages. (c) and (d): An image from EYaleB dataset and its associated subimages.
Figure 2: Parameters of handy and learned Gabor functions depicted in $\mu\nu$-space. The horizontal axis shows the scale ($\nu$) and the vertical axis shows the orientation ($\mu$).

(a) Parameters of the original 40 Gabor functions used by Ren et al. [36].

(b) Parameters of the 40 learned Gabor functions for the CMU-Light dataset.

(c) Parameters of the 40 learned Gabor functions for the EYaleB dataset.
(a) Learned combined kernel for a sample image region from CMU-Light dataset.  
(b) Learned combined kernel for a sample image region from EYaleB dataset.

Figure 3: Compound kernels corresponding to the weighted combination of learned Gabor kernels by the method of Section 3.

(a) Some randomly generated images by combination of two Gabor functions. Centers of Gabor functions are marked with a red plus sign. (b) Non-sparse approximation of images of subfigure (a) with SIKL algorithm. Since the learned Gabor functions are present at all support vector locations, they cannot be assigned to any specific location. (c) An image of support vector pixels in which the darkness of each pixel is proportional to the magnitude of its support vector coefficient. (d) Approximation of images of subfigure (a) obtained by learning a sparse Gabor-based representation. Red/Green plus signs correspond to the centers of the Gabor functions contributing positively/negatively in the learned sparse representation.

Figure 4: (a) Some randomly generated images by combination of two Gabor functions. Centers of Gabor functions are marked with a red plus sign. (b) Non-sparse approximation of images of subfigure (a) with SIKL algorithm. Since the learned Gabor functions are present at all support vector locations, they cannot be assigned to any specific location. (c) An image of support vector pixels in which the darkness of each pixel is proportional to the magnitude of its support vector coefficient. (d) Approximation of images of subfigure (a) obtained by learning a sparse Gabor-based representation. Red/Green plus signs correspond to the centers of the Gabor functions contributing positively/negatively in the learned sparse representation.