Discrete Fourier Transform Based Pattern Classifiers

Stefen Hui* Stanisław H. Żak†

Abstract

A technique for pattern classification using the Fourier transform combined with the nearest neighbor classifier is proposed. The multidimensional fast Fourier transform (FFT) is applied to the patterns in the data base. Then the magnitudes of the Fourier coefficients are sorted in descending order and the first P coefficients with largest magnitudes are selected, where P is a design parameter. These coefficients are then used in further processing rather than the original patterns. When a noisy pattern is presented for classification, the pattern's P Fourier coefficients with largest magnitude are extracted. The coefficients are arranged in a vector in the descending order of their magnitudes. The obtained vector is referred to as the signature vector of the corresponding pattern. Then the distance between the signature vector of the pattern to be classified and the signature vectors of the patterns in the data base are computed and the pattern to be classified is matched with a pattern in the data base whose signature vector is closest to the signature vector of the pattern being classified.

Keywords: Pattern classification, Multidimensional discrete Fourier transform (DFT), Fourier coefficients.

I. Introduction

An essential element of a quality pattern classifier is a feature extraction algorithm that is capable of extracting features that are invariant to certain geometric transformations. In the paper, we focus on classifying images that are transformed from original prototype images by a group of planar transformations (see Section II-A). The Fourier transform possesses a number properties that make it suitable for invariant feature extraction for pattern recognition. Altmann and Reitböck [1] and Reitboeck and Altmann [2] proposed a size- and position-invariant description of an image function via the absolute value of the Mellin transform of its amplitude spectrum (the absolute value of the Fourier transform.) Gardenier, McCallum, and Bates [3] used the Fourier transform amplitudes in pattern recognition applications. More recently, Chen, Bui, and Krzyżak [4] employed the Radon transform and dual-tree complex wavelets, in addition to Fourier transforms, in the invariant pattern recognition.

The method we are proposing uses the amplitude spectra of the images. It is not immediately clear that the amplitude spectrum can uniquely determine the image. In other words, different functions may have the same amplitude spectrum. However, it is well known that functions that arise in practice are uniquely determined by their amplitude spectra, see Barakat and Newsam [5], and Van Hove, Lim, and Oppenheim [6], and Taylor [7]. The study of the determination of a function, either continuous or discrete, from its amplitude spectrum has a long history, see for example, Akutowicz [8], [9], Barakat and Newsam [5], and Van Hove, Lim, and Oppenheim [6]. Many of the studies also discuss the possible recovery of the function from its amplitude spectrum, see for example, Hayes, Lim, and Oppenheim [10], Hayes [11], Taylor [7], and Bates and McDonnell [12].

The feature of each image that we use in our method is the decreasing rearrangement of the amplitude spectrum of the image. There is no reason to believe that the decreasing rearrangement of the amplitude spectrum of an image can uniquely determine the image itself. However, it is our experience that the decreasing rearrangement of the amplitude spectrum does determine the image in all cases we have studied.

We propose an algorithm where P Fourier coefficients with largest absolute values are extracted. The magnitudes of coefficients are arranged in a vector in descending order. We refer to the obtained vector as the signature vector of the corresponding pattern. The distance between the signature vector of the

*Department of Mathematical Sciences, San Diego State University, San Diego, CA 92182, Email: hui@math.sdsu.edu
†School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, Email: zak@purdue.edu

pattern to be classified and the signature vectors of the patterns in the data base are computed and the pattern to be classified is matched with a pattern in the data base whose signature vector is closest to the signature vector of the pattern being classified.

The paper is organized as follows. In Section II, we present relevant background results related to the Fourier transform and its properties that we use to extract pattern features for the purpose of pattern classification. In Section III-B, we discuss the use of discrete Fourier coefficients as signature vectors in noisy environments. In Section IV, we propose a Fourier transform based algorithm for pattern classification. In Section V, we present results of numerical experiments demonstrating the effectiveness of the proposed pattern classifier. Conclusions are found in Section VI.

II. Mathematical Preliminaries

In this section, we collect the mathematical results that we use in our discussion. First, recall that a standard rectangular RGB image is represented by a 3D matrix x of size $M_1 \times M_2 \times 3$. Each of the three $M_1 \times M_2$ submatrices contain the intensity values for red, green, and blue, respectively. We allow the following operations on the images: rotation through $90°, 180°, 270°$, and reflections through the horizontal, median, the vertical median, and the two main diagonals of the image. Let $\mathbf{\Gamma}$ denote the set of these operations. By combining the operations in $\mathbf{\Gamma}$, a total of 28 different images can be generated from a single image. Note that the operations in $\mathbf{\Gamma}$ only affect the first two dimensions of an image. For $\gamma \in \mathbf{\Gamma}$, we use $\gamma(x)$ to denote the image obtained from x using the operation γ.

Let ℓ and $m_1, \ldots, M_\ell$ be positive integers. For our applications, $\ell = 3$ but we present the general case for notational convenience. Let $\boldsymbol{L} \in \mathbb{Z}^\ell$ be the rectangular lattice

$$\boldsymbol{L} = [0, \ldots, M_1 - 1] \times \cdots \times [0, \ldots, M_\ell - 1]$$

and let $\boldsymbol{M} = (M_1, \ldots, M_\ell)$ be the vector containing the dimensions of the lattice $\boldsymbol{L}$. Then the image x can be viewed as a complex-valued function on $\boldsymbol{L}$, that is, $x : \boldsymbol{L} \to \mathbb{C}$.

A. The Discrete Fourier Transform (DFT)

We first recall the following standard notation: Let $\boldsymbol{m} = (m_1, \ldots, m_\ell)$ and $\boldsymbol{M} = (M_1, \ldots, M_\ell)$. We denote the coordinate-wise division of $\boldsymbol{m}$ by $\boldsymbol{M}$ by

$$\frac{\boldsymbol{m}}{\boldsymbol{M}} = \left(\frac{m_1}{M_1}, \ldots, \frac{m_\ell}{M_\ell} \right).$$

The discrete Fourier transform (DFT) of x is defined by

$$\widehat{x}(\boldsymbol{n}) = \sum_{\boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) e^{-2\pi j \, \boldsymbol{n} \cdot \frac{\boldsymbol{m}}{\boldsymbol{M}}} \tag{1}$$

for each $\boldsymbol{n} \in \boldsymbol{L}$. The basic properties of the Fourier transform can be found in [13], [14], [15], [16].

$$e^{-2\pi j \, \boldsymbol{n} \cdot \frac{\boldsymbol{m}}{\boldsymbol{M}}}$$

We make use of the following properties of the DFT. The following theorem is well known:

Theorem 1: Let $x : \boldsymbol{L} \to \mathbb{C}$. Then

$$\|\widehat{x}\|^2 = |\boldsymbol{M}| \|x\|^2,$$

where $|\boldsymbol{M}| = M_1 \cdots M_\ell$.

We also need the following result on the values of $\widehat{x}$. Its proof is elementary but it does not appear in the standard Fourier analysis books and so for the convenience of the reader, we have included its proof. For $x : \boldsymbol{L} \to \mathbb{C}$, let

$$A(x) = \{ |\widehat{x}(\boldsymbol{n})| : \boldsymbol{n} \in \boldsymbol{L} \}$$

denote the set of values of the amplitude spectrum of x.

Theorem 2: For each $x : \boldsymbol{L} \to \mathbb{C}$ and each $\gamma \in \boldsymbol{\Gamma}$,

$$A(x) = A(\gamma(x)).$$

Proof: Suppose x has dimension $M_1 \times M_2 \times \cdots \times M_\ell$. Let $\tau(x)$ denote the vector obtained from x by transposing its first two coordinates. Note that for an image x, $\tau(x)$ is the same as reflecting x through its main diagonal. Then $\tau(x)$ has dimension $M_2 \times M_1 \times \cdots \times M_\ell$ and is a function on the transposed lattice

$$\begin{aligned} \boldsymbol{L}' &= [0, \ldots, M_2 - 1] \times [0, \ldots, M_1 - 1] \times \cdots \times [0, \ldots, M_\ell - 1] \\ &= [0, \ldots, M_2 - 1] \times [0, \ldots, M_1 - 1] \times L_3, \end{aligned}$$

where $\boldsymbol{L}_3 = [0, \ldots, M_3 - 1] \times \cdots \times [0, \ldots, M_\ell - 1]$. We have for $\boldsymbol{n} \in \boldsymbol{L}'$ that

$$\begin{aligned} \widehat{\tau(x)}(\boldsymbol{n}) &= \sum_{\boldsymbol{m}' \in \boldsymbol{L}'} \tau(x)(\boldsymbol{m}') e^{-2\pi j\, \boldsymbol{n} \cdot \frac{\boldsymbol{m}}{\boldsymbol{M}}} \\ &= \sum_{\boldsymbol{m}'' \in \boldsymbol{L}_3} \sum_{m_1'=0}^{M_2-1} \sum_{m_2'=0}^{M_1-1} \tau(x)(m_1', m_2', \boldsymbol{m}'') e^{-2\pi j \left(\frac{m_1' n_1}{M_1} + \frac{m_2' n_2}{M_2} + \boldsymbol{m}'' \cdot \frac{\boldsymbol{n}''}{\boldsymbol{M}} \right)} \\ &= \sum_{\boldsymbol{m}'' \in \boldsymbol{L}_3} \sum_{m_1'=0}^{M_2-1} \sum_{m_2'=0}^{M_1-1} x(m_2', m_1', \boldsymbol{m}'') e^{-2\pi j \left(\frac{m_2' n_2}{M_2} + \frac{m_1' n_1}{M_1} + \boldsymbol{m}'' \cdot \frac{\boldsymbol{n}''}{\boldsymbol{M}} \right)} \\ &= \widehat{x}(\boldsymbol{n}'), \end{aligned}$$

where $\boldsymbol{n}'$ denotes the vector obtained from $\boldsymbol{n}$ by transposing its first two coordinates. We conclude that the DFT commutes with transposition. It follows that

$$A(x) = A(\tau(x)).$$

Let ϕ denote the reflection of the first two coordinates through the horizontal median. Then $\phi(x)$ has the same dimension as x and for $0 \le m_1 \le M_1 - 1$ and $m = (m_1, \boldsymbol{m}_2) \in \boldsymbol{L}$,

$$\phi(x)(m_1, \boldsymbol{m}_2) = x(M_1 - m_1 - 1, \boldsymbol{m}_2).$$

Factor the lattice $\boldsymbol{L}$ into $\boldsymbol{L} = [0, \ldots, M_1 - 1] \times L_2$ and for $\boldsymbol{n} \in \boldsymbol{L}$, let $\boldsymbol{n} = (n_1, \boldsymbol{n}_2)$. Then

$$\begin{aligned} \widehat{\phi(x)}(\boldsymbol{n}) &= \sum_{\boldsymbol{m} \in \boldsymbol{L}} \phi(x)(\boldsymbol{m}) e^{-2\pi j\, \boldsymbol{n} \cdot \frac{\boldsymbol{m}}{\boldsymbol{M}}} \\ &= \sum_{\boldsymbol{m}_2 \in \boldsymbol{L}_2} \sum_{m_1=0}^{M_1-1} \phi(x)(m_1, \boldsymbol{m}_2) e^{-2\pi j \left(\frac{m_1 n_1}{M_1} + \boldsymbol{n}_2 \cdot \frac{\boldsymbol{m}_2}{\boldsymbol{M}_2} \right)} \\ &= \sum_{\boldsymbol{m}_2 \in \boldsymbol{L}_2} \sum_{m_1=0}^{M_1-1} x(M_1 - m_1 - 1, \boldsymbol{m}_2) e^{-2\pi j \left(\frac{m_1 n_1}{M_1} + \boldsymbol{n}_2 \cdot \frac{\boldsymbol{m}_2}{\boldsymbol{M}_2} \right)} \\ &= \sum_{\boldsymbol{m}_2 \in \boldsymbol{L}_2} \sum_{m=0}^{M_1-1} x(m, \boldsymbol{m}_2) e^{-2\pi j \left(\frac{(M_1 - m - 1) n_1}{M_1} + \boldsymbol{n}_2 \cdot \frac{\boldsymbol{m}_2}{\boldsymbol{M}_2} \right)} \\ &= e^{-2\pi j \frac{(M_1 - 1) n_1}{M_1}} \sum_{\boldsymbol{m}_2 \in \boldsymbol{L}_2} \sum_{m=0}^{M_1-1} x(m, \boldsymbol{m}_2) e^{-2\pi j \left(-\frac{m n_1}{M_1} + \boldsymbol{n}_2 \cdot \frac{\boldsymbol{m}_2}{\boldsymbol{M}_2} \right)} \\ &= e^{2\pi j \frac{n_1}{M_1}} \sum_{\boldsymbol{m}_2 \in \boldsymbol{L}_2} \sum_{m=0}^{M_1-1} x(m, \boldsymbol{m}_2) e^{-2\pi j \left(\frac{m (M_1 - n_1)}{M_1} + \boldsymbol{n}_2 \cdot \frac{\boldsymbol{m}_2}{\boldsymbol{M}_2} \right)} \\ &= e^{2\pi j \frac{n_1}{M_1}} \widehat{x}(M_1 - n_1, \boldsymbol{n}_2). \end{aligned}$$

The next to last equality is obtained using the fact that $e^{2\pi j k} = 1$ for all integers k. It follows that for $\boldsymbol{n} = (n_1, \boldsymbol{n}_2) \in \boldsymbol{L}$,

$$|\widehat{\phi(x)}(\boldsymbol{n})| = \begin{cases} |\widehat{x}(0, \boldsymbol{n}_2)| & \text{if } n_1 = 0 \\ |\widehat{x}(M_1 - n_1, \boldsymbol{n}_2)| & \text{if } 1 \le n_1 \le M_1 - 1. \end{cases}$$

Thus the magnitude of the DFT commutes with reflection through the horizontal median and so $A(x) = A(\phi(x))$. The other operations in Γ can all be expressed as combinations of τ and ϕ:

1) ρ = Counter-clockwise rotation by $90°$ = $\phi \circ \tau$
2) ν = Reflection through vertical median = $\rho^3 \circ \phi \circ \rho$
3) Reflection through opposite diagonal = $\nu \circ \tau \circ \nu$

It follows that $A(x)$ is invariant under operations in Γ. ■

Theorem 2 shows that the magnitude of the DFT is invariant under the operations in Γ. This property makes the magnitude of the DFT an attractive tool in the design of recognition algorithms that are robust against the operations in Γ. However, we cannot use the magnitude of the DFT directly as it is not invariant under the operations in Γ; only the unordered values are. We show in Section III-B that there are drawbacks to using the entire DFT in noisy environments.

B. Rearrangement of Vectors

In our algorithm, we arrange the values of $A(x)$ in decreasing order of magnitude. In this section, we state and prove two simple results on rearranged vectors. The results are classical and we include their proofs since they do not seem to be widely known outside of classical analysis.

Lemma 3: Suppose $\boldsymbol{a}$ and $\boldsymbol{b}$ are real vectors of length N. Then the maximum of the dot products of all possible rearrangements of $\boldsymbol{a}$ and $\boldsymbol{b}$ is achieved when the values in each vector are arranged in decreasing order.

Proof: Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be real vectors of length N. If $a_1 = \cdots = a_N$, then the claim is certainly true because all rearrangements of $\boldsymbol{b}$ will give the same dot product. We can assume without loss of generality that $a_1 \geq a_2 \geq \cdots \geq a_N$. and that not all a_k's are the same. Let s be the maximum of the dot products. Suppose

$$s = a_1 b_1' + \cdots + a_N b_N'$$

for an rearrangement $(b_1', \ldots, b_N')$ of $(b_1, \ldots, b_N)$. Suppose $j < k$ and $a_j > a_k$. Then we must have $b_j' \geq b_k'$ because otherwise we have

$$a_j b_k' + a_k b_j' - (a_j b_j' + a_k b_k') = (a_j - a_k)(b_k' - b_j') > 0$$

and we can obtain a larger dot product by switching b_j' and b_k'. The claim now follows by rearranging the b_k's in a range where the a_k's are constant in decreasing order. ■

The following is now immediate.

Theorem 4: Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be real vectors of length N and let $\boldsymbol{a}^\sharp$ and $\boldsymbol{b}^\sharp$ be the rearrangements of $\boldsymbol{a}$ and $\boldsymbol{b}$ in decreasing order. Then

$$\|\boldsymbol{a}^\sharp - \boldsymbol{b}^\sharp\| \leq \|\boldsymbol{a} - \boldsymbol{b}\|.$$

Proof: Since $\|\boldsymbol{a}^\sharp\|^2 = \|\boldsymbol{a}\|^2$, $\|\boldsymbol{b}^\sharp\|^2 = \|\boldsymbol{b}\|^2$, and $\boldsymbol{a}^\sharp \cdot \boldsymbol{b}^\sharp \geq \boldsymbol{a} \cdot \boldsymbol{b}$, we have

$$\begin{aligned} \|\boldsymbol{a}^\sharp - \boldsymbol{b}^\sharp\|^2 &= \|\boldsymbol{a}^\sharp\|^2 - \boldsymbol{a}^\sharp \cdot \boldsymbol{b}^\sharp + \|\boldsymbol{b}^\sharp\|^2 \\ &\leq \|\boldsymbol{a}\|^2 - \boldsymbol{a} \cdot \boldsymbol{b} + \|\boldsymbol{b}\|^2 \\ &= \|\boldsymbol{a} - \boldsymbol{b}\|. \end{aligned}$$

■

III. The Pattern Recognition Algorithm

In this section, we give the motivation for the algorithm based on the mathematical results given in Section II and the impact of noise (see Section III-B).

A. Motivation

Let $x_1, \dots, x_N : \boldsymbol{L} \to \mathbb{R}$ be the pixel values of N distinct ℓ-dimensional prototype images. The original image for a received noiseless y that has not been rotated or reflected can be recovered exactly by solving

$$x_{est} = \arg \min_{k=1,\dots,N} \|y - x_k\|_2. \tag{2}$$

However, if the image has been rotated or reflected, then this method does not work. One approach would be to compare the received image to all possible images obtainable from the original set of prototype images. While this can be done, it increases the computational load almost 30 fold, which makes this approach less desirable.

Since the DFT is not invariant under rotation and reflection, using the DFT directly suffers the same problem as using the original image values. However, as we proved in Theorem 2, the set of values of the amplitude spectrum

$$A(x) = \{|\widehat{x}(\boldsymbol{m})| : \boldsymbol{m} \in \boldsymbol{L}\}$$

is invariant under operations in Γ. So we need to find a metric on sets that measures the difference between $A(x)$ and $A(y)$, and then use that to measure the difference between the images x and y. The method we chose is based on Theorem 4. Let $A^\sharp(x)$ denote the vector obtained from $A(x)$ by decreasing rearrangement. From Theorem 4 and Theorem 1, we have

$$\|A^\sharp(x) - A^\sharp(y)\| \le \|\widehat{x} - \widehat{y}\| = \sqrt{|\boldsymbol{M}|}\|x - y\|. \tag{3}$$

Let

$$d(x, y) = \|A^\sharp(x) - A^\sharp(y)\|. \tag{4}$$

Then it is easy to see that d is a semi-metric on the set of prototype images but it is not a metric on the set of all possible images. It is not a metric because it is possible to have $A^\sharp(x) = A^\sharp(y)$ even if $x \ne y$, but this rarely happens in practice for a finite set of images.

Remark 1: We can formalize the way images can be distinguished using the following equivalence relation. We say that the images x and y are equivalent, denoted by $x \sim y$, if $A(x) = A(y)$. It can easily be verified that $\sim$ is an equivalence relation on the space $\mathcal{I}$ of images with the same number of pixels. For each image x, let $[x]$ denote the equivalence class containing x. On the space $\mathcal{I}/\sim$ of equivalence classes, let

$$D([x], [y]) = d(x, y), \tag{5}$$

where d is the semi-metric defined in equation (4). Then D is a metric on $\mathcal{I}/\sim$.

We will assume that our prototype images have distinct equivalence classes. Then for $x \ne y$, $d(x, y) = D([x], [y]) > 0$ and so d is a metric on the set of prototype images.

For a received image y, $d(x, y) > 0$ if $x \ne y$ and the only solution of

$$x_{est} = \arg \min_{k=1,\dots,N} d(x_k, y). \tag{6}$$

is $x = y$, the original message. We have thus found a method for recovering noiseless images that have been transformed by the operations in Γ.

We next discuss the case when there is noise.

B. Noise Considerations

Let $x_1, \ldots, x_N : \boldsymbol{L} \to \mathbb{R}$ be the pixel values of N ℓ-dimensional prototype images. Note that standard color images are three-dimensional. The noisy version of the k-th image has the form

$$y_k = x_k + \omega_k, \tag{7}$$

where ω_k a random variable with zero mean and finite variance σ^2 per pixel. Furthermore, we assume that $\{\omega_k(\boldsymbol{m}) : k = 1, \ldots, N,\ \boldsymbol{m} \in \boldsymbol{L}\}$ are independent and identically distributed (iid) random variables. The maximum likelihood estimator for a received noisy image y is

$$x_{est} = \arg \min_{k=1,\ldots,N} \|y - x_k\|_2. \tag{8}$$

However, as we had seen before, the above estimator is not robust against rotation or translation and we chose instead to use

$$x_{est} = \arg \min_{k=1,\ldots,N} d(x_k, y), \tag{9}$$

where $d(x, y) = \|A^\sharp(x) - A^\sharp(y)\|$. If the noise is zero, then the actual image is a solution to the above optimization problem, and would be the unique solution when d is a metric. However, as we next demonstrate, there is drawback to this approach in a noisy environment.

Let x be a prototype image and $y = x + \omega$ a noisy version of the image. The $\boldsymbol{n}$-th Fourier coefficient of y satisfies

$$\begin{aligned}
|\widehat{y}(\boldsymbol{n})|^2 &= \left| \sum_{\boldsymbol{m} \in \boldsymbol{L}} (x(\boldsymbol{m}) + \omega(\boldsymbol{m})) e^{-2\pi j\, \boldsymbol{n} \cdot \frac{\boldsymbol{m}}{\boldsymbol{M}}} \right|^2 \\
&= \sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) \overline{x(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \\
&\quad + 2 \text{ Re } \left(\sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \right) \\
&\quad + \sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} \omega(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \\
&= |\widehat{x}(\boldsymbol{n})|^2 + 2 \text{ Re } \left(\sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \right) \\
&\quad + \sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} \omega(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}}
\end{aligned}$$

The expected value of $|\widehat{y}(\boldsymbol{n})|^2$ is

$$\begin{aligned}
\boldsymbol{E}\left[|\widehat{y}(\boldsymbol{n})|^2\right] &= |\widehat{x}(\boldsymbol{n})|^2 \\
&\quad + \boldsymbol{E}\Bigg[2 \text{ Re } \left(\sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \right) \\
&\qquad + \sum_{\boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{L}} \omega(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})} e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \Bigg]
\end{aligned}$$

Since $\omega(\boldsymbol{m})$ and $\omega(\boldsymbol{k})$ are independent with mean zero and variance σ^2, we have

$$\begin{aligned}
\boldsymbol{E}\left[|\widehat{y}(\boldsymbol{n})|^2\right] &= |\widehat{x}(\boldsymbol{n})|^2 \\
&\quad + \sum_{\boldsymbol{k} = \boldsymbol{m} \in \boldsymbol{L}} \boldsymbol{E}\left[\omega(\boldsymbol{m}) \overline{\omega(\boldsymbol{k})}\right] e^{-2\pi j\, (\boldsymbol{m}-\boldsymbol{k}) \cdot \frac{\boldsymbol{n}}{\boldsymbol{M}}} \\
&= |\widehat{x}(\boldsymbol{n})|^2 + |\boldsymbol{M}| \sigma^2,
\end{aligned} \tag{10}$$

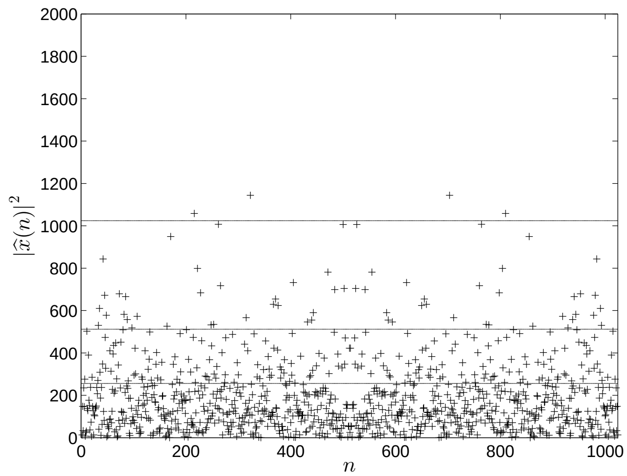


Fig. 1. A plot of the magnitude squared of the Fourier coefficients in Example 1.

where $|\boldsymbol{M}| = M_1 \cdots M_\ell$ is the number of pixels in the image. In light of equation (10), we see that a particular Fourier coefficient $\widehat{x}(\boldsymbol{n})$ is useful as a feature only if $|\widehat{x}(\boldsymbol{n})|^2$ is significantly greater than $M\sigma^2$. For most practical images with a moderate number of pixels, the number of Fourier coefficients whose squared-amplitudes are greater than two or three times $M\sigma^2$ is quite small for any nontrivial noise variance σ^2. Note this observation comports with the Riemann-Lebesgue lemma, see for example [17, p. 195]. We illustrate this crucial point with a numerical example.

Example 1: We generate a random zero-one vector x of length 1024 where each entry has probability 0.25 of being one. We compute the Fourier coefficients of x and plot $|\widehat{x}(n)|^2$ as a function of n. The three horizontal lines are at height 0.25×1024, 0.5×1024, and 1024, respectively. The plot was generated by the following simple Matlab script:

```
x=rand([1 1024])>0.75;
x_hat=fft(x);
plot(abs(x_hat.^2),'r+')
axis([0 1024 0 2*10^3])
hold on
plot(1:1024,1024*0.25)
plot(1:1024,1024*0.5)
plot(1:1024,1024*1)
hold off
```

Note that the plot excludes the high DC component at frequency 0. This is accomplished in the above script using the command `axis([0 1024 0 2*10^3])`.

The significance of equation (10) is that it implies that most of Fourier coefficients are not useful as features of an image in a noisy environment. In fact, the difference $d(x_k, y)$ in equation (9) may be overwhelmed by noise. It therefore makes sense to only consider the Fourier coefficients with the largest magnitudes. Note that using a scaled version of the DFT, for example,

$$\widehat{x}(\boldsymbol{n}) = \frac{1}{M} \sum_{\boldsymbol{m} \in \boldsymbol{L}} x(\boldsymbol{m}) e^{-\frac{2\pi j\, \boldsymbol{m} \cdot \boldsymbol{n}}{M}},$$

does not alleviate the problem because all coefficients are scaled the same and so the signal-to-noise ratio does not change.

In light of the above discussion, we use only the largest values in $A^\sharp(x)$. Let $A^\sharp_P(x)$ be the vector containing the first P coordinates of $A^\sharp(x)$ and let $d_P(x,y) = \|A^\sharp_p(x) - A^\sharp_P(y)\|$. In all practical situations, d_P is a metric on the set of prototype images and we solve

$$x_{est} = \arg \min_{k=1,\ldots,N} d_P(x_k, y) \tag{11}$$

to find the most likely original image given the received image y.

C. Error Analysis

In this section, we analyze the effect of noise on the proposed algorithm. Our main tool is Equation 3. As in Section III-B, we assume that the noisy version of the k-th image has the form

$$y_k = x_k + \omega_k, \tag{12}$$

where ω_k a random variable with zero mean and finite variance σ^2 per pixel. We have from Equation (3) that

$$d^2(x, x+\omega) = \|A^\sharp(x) - A^\sharp(x+\omega)\|^2 \leq \|\widehat{\omega}\|^2 = |\boldsymbol{M}|\|\omega\|^2$$

and it follows that the expected value of the squared error for the whole image satisfies

$$\begin{aligned} E\left[\|A^\sharp(x) - A^\sharp(x+\omega)\|^2\right] &\leq |\boldsymbol{M}| E\left[\|\omega\|^2\right] \\ &= |\boldsymbol{M}| \sum_{\boldsymbol{m}\in\boldsymbol{L}} E\left[|\omega(\boldsymbol{m})|^2\right] \\ &= |\boldsymbol{M}| \sum_{\boldsymbol{m}\in\boldsymbol{L}} \sigma^2 \\ &= |\boldsymbol{M}|^2\sigma^2. \end{aligned} \tag{13}$$

In the above computation, we used the fact that $\sigma^2 = E\left[|\omega(\boldsymbol{m})|^2\right]$ is the noise variance for each pixel. Since there are $|\boldsymbol{M}|$ pixels in each image, the expected squared error per pixel is

$$\frac{1}{|\boldsymbol{M}|} E[\|A^\sharp(x) - A^\sharp(x+\omega)\|^2] = |\boldsymbol{M}|\sigma^2.$$

When compared with the un-rearranged per pixel error given in Equation (10), we can see that rearrangement does not increase the per pixel mean-square error of the DFT.

We have for a fixed image x and a noise sample ω that

$$\|A^\sharp(x) - A^\sharp(x+\omega)\|^2 \leq |\boldsymbol{M}|\|\omega\|^2 = |\boldsymbol{M}|^2 \left(\frac{1}{|\boldsymbol{M}|} \sum_{\boldsymbol{m}\in\boldsymbol{L}} |\omega(\boldsymbol{m})|^2 \right).$$

IV. Pattern Recognition Algorithm

The pattern recognition method we propose uses P leading elements of the decreasing rearrangements of the magnitude of the Fourier coefficients of the images as signatures. The number P is a design parameter chosen on the basis of nature of the prototype images and the expected range of noise variance. The complete algorithm can be summarized as follows:

1) Prototype Image Signature Extraction
   a) Fix a design parameter P.
   b) For each image given by a function x defined on a lattice $\boldsymbol{L}$, compute its discrete Fourier transform $\widehat{x}$.
   c) Evaluate and sort $\{|\widehat{x}(\boldsymbol{n})|$: $\boldsymbol{n} \in \boldsymbol{L}\}$ in descending order.
   d) Store the P highest values in a vector as the signature for the image.
2) Noisy Image Pattern Classification
   a) For a noisy image y to be classified, repeat steps 1b, 1c, and 1d described above.
   b) From the stored prototype signature vectors, find the one with smallest distance from the signature vector of y.

Note that one can use any metric on $\mathbb{R}^P$ as the distance function in step 2b. In our simulations, we use the ℓ^1 norm, which gives almost the same performance as the ℓ^2 norm.

TABLE I
NOISE STANDARD DEVIATIONS USED IN SIMULATION.

Noise Std σ	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.50	3.00	3.50	4.00	4.50	5.00

V. SIMULATIONS

For our simulations, we used 14 color images of stamps from [18] that contain images of butterflies. Each image contains 300×200 color pixels, which we treat as a function on a $300 \times 200 \times 3$ integer lattice. The original pixel values were integers from 0 to 255, which we converted into floating point real numbers with values between 0 and 1. We computed the three dimensional discrete Fourier transform of each image and then sorted the magnitudes of the Fourier coefficients. We kept the top 30 magnitudes from each image as its signature.

For the simulations, one of the 14 prototype images was randomly and uniformly selected. Gaussian noise was added independently to each pixel to the selected image; see Figure 2 for an illustration of a typical prototype image and its noisy version for selected noise levels. We use the standard definition of signal-to-noise ratio (SNR) for an image x:

$$\text{SNR} = \sqrt{\frac{1}{|\boldsymbol{M}|\sigma^2} \sum_{\boldsymbol{n} \in \boldsymbol{L}} [x(\boldsymbol{n}) - \bar{x}]^2}$$

where σ^2 is the variance of the noise per pixel and $\bar{x} = \sum_{\boldsymbol{n} \in \boldsymbol{L}} x(\boldsymbol{n})/M$ is the average pixel value of the image x; see for example Chen, Bui, and Krzyzak [4]. Note that the above definition of SNR is independent of image size and can be interpreted as the image SNR and as the average per pixel SNR.

The noisy image was rotated randomly and uniformly by $0°$, $90°$, $180°$, or $270°$; see Figure 3 for an illustration of the rotations of a typical image. The signature vector of the noisy rotated image was extracted and the prototype image with the closest signature vector in ℓ^1 norm was selected as the best estimate. The number of errors was recorded. Note that there are a total of 56 possible images without noise.

We repeated the same experiment using compressed versions of the same images. Each of three color components of an image was compressed by a two dimensional Daubechies wavelet to have size 75×50. The three compressed color components combine to form a compressed color image of size $75 \times 50 \times 3$; see Figure 4 for a typical compressed image and selected noisy versions.

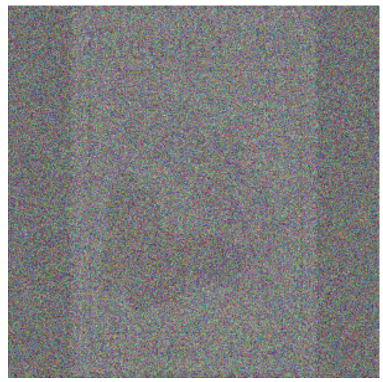
We chose the noise standard deviations given in Table I for our simulation. For each noise standard deviation, $12,000$ trials were conducted and the number of errors recorded. The results are presented in Figures 5, 6, and 7. In Figure 5, the error rate, which is the number of errors divided by the number of trials, of the simulation using the original prototype images is plotted against the SNR. In Figure 6, we plot the error rate of the simulation using the compressed images. A comparison of the error rates for the original and compressed images is given in Figure 7. In the plots, the SNR is defined using the average signal standard deviation $\sigma = 0.45$ of all prototype images.

Remark 2: We also performed the same experiments with only rotation but no noise. The algorithm was able to identify all presented images correctly and there were no errors. This is exactly as the theory predicted and this also shows that d_P is indeed a metric on the set of prototype images.

(a) Sample image

(b) SNR $= -0.5\,\mathrm{dB}$

(c) SNR $= -3.5\,\mathrm{dB}$



(d) SNR $= -6.5\,\mathrm{dB}$

Fig. 2. Original image and its noisy versions used in the numerical experiment.

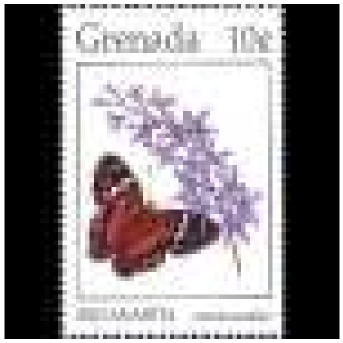
(a) Sample image

(b) 90°

(c) 180°

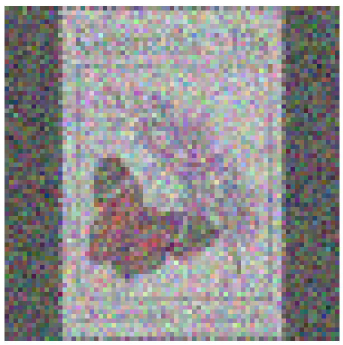
(d) 270°

Fig. 3. Original image and its rotated versions used in the numerical experiment.

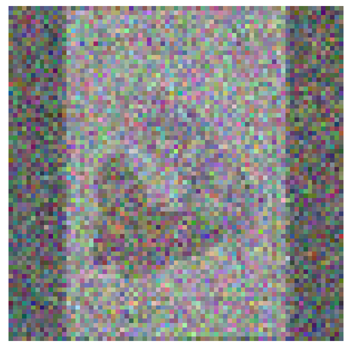


(a) Sample image

(b) SNR $= -0.5$ dB



(c) SNR $= -3.5$ dB



(d) SNR $= -6.5$ dB

Fig. 4. A compressed image and its noisy versions.

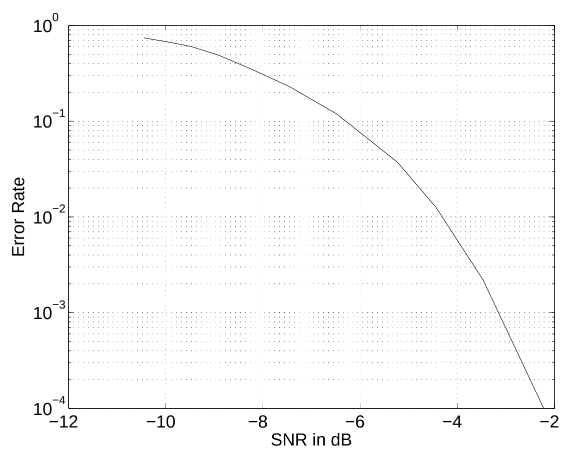


Fig. 5. A plot of the error rate versus signal-to-noise ratio - original images.

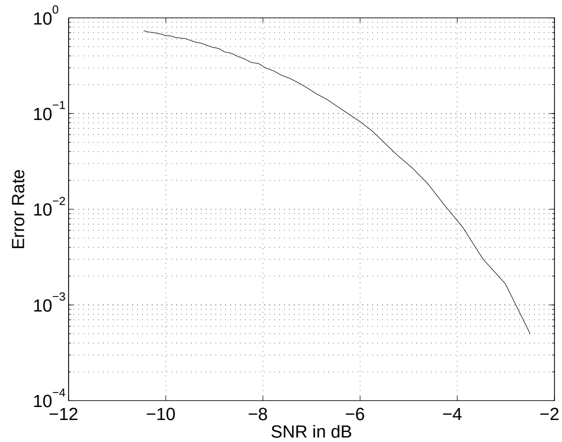


Fig. 6. Error rate versus signal-to-noise ratio - compressed images.

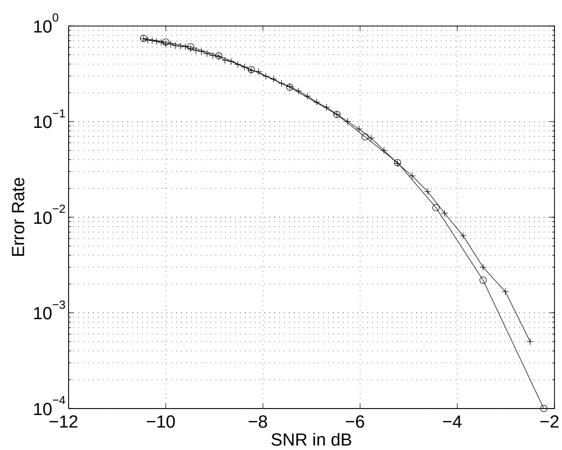


Fig. 7. Error rate versus signal-to-noise ratio - comparison for the compressed images and uncompressed images.

VI. Conclusions

We presented a Fourier transform based pattern recognition algorithm. The algorithm uses the magnitudes of a small number of Fourier coefficients of an image as its signature vector. We presented mathematical justifications as to why using a small number of Fourier coefficients as a signature vector may be better than using all Fourier coefficients in the presence of noise. Simulations were conducted to demonstrate the effectiveness of the algorithm in noisy environments.

References

[1] J. Altmann and H. J. P. Reitböck, "A fast correlation method for scale- and translation-invariant pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 1, pp. 46–57, January 1984.

[2] H. J. Reitboeck and J. Altmann, "A model for size- and rotation-invariant pattern processing in the visual system," *Biological Cybernetics*, vol. 51, no. 2, pp. 113–121, 1984.

[3] P. H. Gardenier, B. C. McCallum, and R. T. Bates, "Fourier transform magnitudes are unique pattern recognition templates," *Biological Cybernetics*, vol. 54, no. 6, pp. 385–391, June 1986.

[4] G. Y. Chen, T. D. Bui, and A. Krzyżak, "Invariant pattern recognition using radon, dual-tree complex wavelet and Fourier transforms," *Pattern Recognition*, vol. 42, no. 9, pp. 2013–2019, 2009.

[5] R. Barakat and G. Newsam, "Necessary conditions for a unique solution to two-dimensional phase recovery," *J. of Mathematical Physics*, vol. 25, no. 11, pp. 3190–3193, November 1984.

[6] P. L. Van Hove, J. S. Lam, and A. V. Oppenheim, "Signal reconstruction from Fourier transform amplitude," in *Applications of Digital Image Processing IV*, A. G. Tescher, Ed. P.O. Box 10, Bellingham, Washington 98227-0010: SPIE—The International Society for Optical Engineering, 1982, pp. 214–225.

[7] L. S. Taylor, "The phase retrieval problem," *IEEE Transactions on Antennas and Propagation*, vol. AP-29, no. 2, pp. 386–391, March 1981.

[8] E. J. Akutowicz, "On the determination of the phase of a fourier integral, I," *Transactions of the American Mathematical Society*, vol. 83, pp. 179–192, 1956.

[9] ——, "On the determination of the phase of a fourier integral, II," *Proceedings of the American Mathematical Society*, vol. 8, pp. 234–238, 1957.

[10] M. H. Hayes, J. S. Lim, and A. V. Oppenheim, "Signal reconstruction from phase or magnitude," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 6, pp. 672–680, December 1980.

[11] M. H. Hayes, "The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, no. 2, pp. 140–154, April 1982.

[12] R. H. T. Bates and M. J. McDonnell, *Image Restoration and Reconstruction*. New York: Oxford University Press, 1986.

[13] H. Dym and H. P. McKean, *Fourier Series and Integrals*. New York: Academic Press, 1972.

[14] E. M. Stein and G. L. Weiss, *Introduction to Fourier Analysis on Euclidean Spaces*. Princeton, N.J.: Princeton University Press, 1971.

[15] J. S. Walker, *Fourier Analysis*. New York: Oxford University Press, 1988.

[16] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1977.

[17] L. Debnath and P. Mikusiński, *Introduction to Hilbert Spaces with Applications*. San Diego: Academic Press, Inc., 1990.

[18] C. Oh and S. H. Żak, "Image recall using a large scale generalized Brain-State-in-a-Box neural network," *International Journal of Applied Mathematics and Computer Science*, vol. 15, no. 1, pp. 99–114, 2005.