Stationarity

1 Overview

The topic of this chapter is stationarity, which is a fundamental concept in signal processing, and in applications of machine learning to sound and images. Imagine that you want to fit a linear model to perform an image-processing task. For example, the model functions a noisy image to a clean image, a problem commonly known as denoising. A linear model estimates each pixel as a linear combination of all other pixels. If the images have N pixels, then fitting the model amounts to estimating an $N \times N$ matrix. Even pretty small images have 10^4 pixels; the corresponding linear model would have 100 million parameters! Learning an arbitrary linear model is therefore completely impractical, in terms of computational and memory resources and in terms of the data needed to fit the model. In order to alleviate this issue, models in signal processing often assume that signals are *stationary*, which means that they have translation-invariant statistics. Nonlinear models based on this idea (convolutional neural networks) are the current state of the art for most image-processing tasks.

Section 2 describes the effect of translations on the frequency representation of a signal. Section 3 introduces linear translation-invariant models, and the convolution operation. Section 4 defines stationary signals and shows that their principal directions are sinusoidal. Section 5 describes linear estimation under stationarity assumptions, which is commonly known as Wiener filtering. Section 6 introduces convolutional neural networks, which are nonlinear stationary models, and shows their remarkable performance for image denoising.

2 Translation

We consider circular translations, where the signals are interpreted as being periodic with period N so the translations *wrap around*. We index all vectors from 0 to $N-1$, i.e. the first entry of a vector x is $x[0]$, the second $x[1]$, and so on. Extending our results to translations that are not circular requires being careful about border effects.

Definition 2.1 (Circular translation)**.** *We denote by $x^{\downarrow s}$ the sth circular translation or shift of a vector $x \in \mathbb{C}^N$. For all $0 \leq j \leq N-1$,*

$$x^{\downarrow s}[j] = x[(j-s) \bmod N]. \tag{1}$$

For an $N \times N$ signal $X \in \mathbb{C}^{N \times N}$, circular translation by $(s_1, s_2) \in \mathbb{C}^{N \times N}$ is denoted by $X^{\downarrow (s_1,s_2)}$. For all $0 \leq j_1, j_2 \leq N-1$,

$$X^{\downarrow (s_1,s_2)}[j_1, j_2] = X[(j_1 - s_1) \bmod N, (j_2 - s_2) \bmod N]. \tag{2}$$

Recall that we use complex exponentials to represent sinusoids when computing frequency representations, because the phase can be encoded as a linear coefficient. Shifting a sinusoid amounts to modifying its phase. As a result, shifting a complex sinusoid is equivalent to multiplying it by a complex constant.

Lemma 2.2. *For any discrete complex sinusoid $\psi_k \in \mathbb{C}^N$ with integer frequency k and any integer shift s*

$$\psi_k^{\downarrow s} = \exp\left(-\frac{i2\pi ks}{N}\right)\psi_k. \tag{3}$$

Proof. By the definition of a complex sinusoid,

$$\psi_k^{\downarrow s}[l] = \exp\left(\frac{i2\pi k(l-s)}{N}\right) \tag{4}$$

$$= \exp\left(-\frac{i2\pi ks}{N}\right)\psi_k[l]. \tag{5}$$

□

Corollary 2.3. *The discrete complex sinusoid $\psi^{\text{2D}}_{k_1,k_2} \in \mathbb{C}^{N\times N}$ with integer frequencies k_1 and k_2*

$$(\psi^{\text{2D}}_{k_1,k_2})^{\downarrow(s_1,s_2)} = \exp\left(-\frac{i2\pi k_1 s_1}{N}\right)\exp\left(-\frac{i2\pi k_2 s_2}{N}\right)\psi^{\text{2D}}_{k_1,k_2}. \tag{6}$$

Proof. The result follows immediately from Lemma 2.2 because $(\psi^{\text{2D}}_{k_1,k_2})^{\downarrow(s_1,s_2)} = \psi_{k_1}^{\downarrow s_1}(\psi_{k_2}^{\downarrow s_2})^T$. □

We can express any vector as a sum of discrete sinusoids each scaled by the corresponding DFT coefficient. If the vector is shifted, then Lemma 2.2 implies that each discrete sinusoid is just scaled by a complex constant. This yields a simple expression for the Fourier coefficients of shifted signals.

Theorem 2.4. *Let $x \in \mathbb{C}^N$ with DFT $\hat{x}$ and $y := x^{\downarrow s}$, $0 \le s \le N-1$. The DFT coefficients of y are given by*

$$\hat{y}[k] := \exp\left(-\frac{i2\pi ks}{N}\right)\hat{x}[k], \quad 0 \le k \le N-1. \tag{7}$$

Let $X \in \mathbb{C}^{N\times N}$ with DFT $\widehat{X}$ and $Y := X^{\downarrow(s_1,s_2)}$, $0 \le s_1, s_2 \le N-1$. The DFT coefficients of Y are given by

$$\widehat{Y}[k_1,k_2] := \exp\left(-\frac{i2\pi ks_1}{N}\right)\exp\left(-\frac{i2\pi ks_2}{N}\right)\widehat{X}[k_1,k_2], \quad 1 \le k_1, k_2 \le N. \tag{8}$$

Proof. We only prove the one-dimensional case, the 2D case follows by the same argument:

$$\hat{y}[k] := \langle x^{\downarrow s}, \psi_k \rangle \tag{9}$$

$$= \langle x, \psi_k^{\downarrow -s} \rangle \tag{10}$$

$$= \left\langle x, \exp\left(\frac{i2\pi ks}{N}\right) \psi_k \right\rangle \quad \text{by Lemma 2.2} \tag{11}$$

$$= \exp\left(-\frac{i2\pi ks}{N}\right) \langle x, \psi_k \rangle \tag{12}$$

$$= \exp\left(-\frac{i2\pi ks}{N}\right) \hat{x}[k]. \tag{13}$$

□

3 Linear translation-invariant models

As explained in Section 1, fitting linear models to perform signal processing tasks is often intractable. Here we introduce *translation-invariant* linear models. Such models have the following property: if we shift their input, then the corresponding output is also shifted, but otherwise stays the same.

Definition 3.1 (Linear translation-invariant function). *A function $\mathcal{F}$ from $\mathbb{C}^N$ to $\mathbb{C}^N$ is linear if for any vectors $x, y \in \mathbb{C}^N$ and any constant $\alpha \in \mathbb{C}$*

$$\mathcal{F}(x + y) = \mathcal{F}(x) + \mathcal{F}(y), \tag{14}$$

$$\mathcal{F}(\alpha x) = \alpha \mathcal{F}(x), \tag{15}$$

and translation invariant if for any shift $0 \leq s \leq N - 1$

$$\mathcal{F}(x^{\downarrow s}) = \mathcal{F}(x)^{\downarrow s}. \tag{16}$$

A function $\mathcal{F}$ from $\mathbb{C}^{N\times N}$ to $\mathbb{C}^{N\times N}$ is linear if for any $X, Y \in \mathbb{C}^{N\times N}$ and any constant $\alpha \in \mathbb{C}$

$$\mathcal{F}(X + Y) = \mathcal{F}(X) + \mathcal{F}(Y), \tag{17}$$

$$\mathcal{F}(\alpha X) = \alpha \mathcal{F}(X), \tag{18}$$

and translation invariant if for any $0 \leq s_1, s_2 \leq N - 1$

$$\mathcal{F}(X^{\downarrow(s_1,s_2)}) = \mathcal{F}(X)^{\downarrow(s_1,s_2)}. \tag{19}$$

Recall that any finite-dimensional linear function can be represented by a matrix. Let $\mathcal{F}_L : \mathbb{C}^N \to \mathbb{C}^N$ be a linear function. We have

$$\mathcal{F}_L(x) = \mathcal{F}_L\left(\sum_{j=0}^{N-1} x[j] e_j\right) \tag{20}$$

$$= \sum_{j=0}^{N-1} x[j] \mathcal{F}_L(e_j) \tag{21}$$

$$= \begin{bmatrix} \mathcal{F}_L(e_0) & \mathcal{F}_L(e_1) & \cdots & \mathcal{F}_L(e_{N-1}) \end{bmatrix} x \tag{22}$$

$$= Mx \tag{23}$$

for any $x \in \mathbb{C}^N$, where e_j is the jth standard vector ($e_j[j] = 1$ and $e_j[k] = 0$ for $k \neq j$). The matrix M is N by N, its jth column is equal to the output corresponding to the jth standard-basis vector.

Now let us consider a linear translation-invariant (LTI) function $\mathcal{F}$. Since $e_j = e_0^{\downarrow j}$ the columns of the corresponding matrix are just shifts of each other. Consequently, the whole function can be represented completely by its action on the first standard-basis vector e_0:

$$\mathcal{F}(x) = \sum_{j=0}^{N-1} x[j]\mathcal{F}(e_j) \tag{24}$$

$$= \sum_{j=0}^{N-1} x[j]\mathcal{F}\left(e_0^{\downarrow j}\right) \tag{25}$$

$$= \sum_{j=0}^{N-1} x[j]\mathcal{F}(e_0)^{\downarrow j}. \tag{26}$$

Standard-basis vectors are often called *impulses* in the signal-processing literature, so the corresponding output of the function is called an impulse response.

Definition 3.2 (Impulse response)**.** *Let $e_0 \in \mathbb{C}^N$ be the first standard-basis vector if we index the entries from 0 to $N-1$ ($e_0[0] = 1$ and $e_0[j] = 0$ for $0 < j \leq N-1$). The vector $h_{\mathcal{F}} \in \mathbb{C}^n$ obtained by applying a function $\mathcal{F} : \mathbb{C}^n \to \mathbb{C}^n$ to e_0 is called the impulse response of the function,*

$$h_{\mathcal{F}} := \mathcal{F}(e_0). \tag{27}$$

Let $E_0 \in \mathbb{C}^{N\times N}$ be the first standard-basis vector in $\mathbb{C}^{N\times N}$ if we index each dimension from 0 to $N-1$ ($E[0,0] = 1$ and $E[j_1, j_2] = 0$ for $0 < j_1, j_2 \leq N-1$). The vector $H_{\mathcal{F}} \in \mathbb{C}^{N\times N}$ obtained by applying a function $\mathcal{F} : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times N}$ to E_0 is the impulse response of the function,

$$H_{\mathcal{F}} := \mathcal{F}(E_0). \tag{28}$$

By Eq. (26) the action of an LTI function on any vector can be expressed in terms of the impulse response

$$\mathcal{F}(x) = \sum_{j=0}^{N-1} x[j]h_{\mathcal{F}}^{\downarrow j}. \tag{29}$$

An LTI function can therefore be parametrized by its impulse response, which has dimension N. This is a dramatic reduction from the N^2 parameters needed to store an arbitrary linear function. The operation described by Eq. (29) is equal to a sum of shifted copies of the impulse response $h_{\mathcal{F}}$ weighted by the entries of the signal x. This is known as a *convolution*. Note that we index the entries of the vectors from 0 to $N-1$, but this is without loss of generality, the definition is equivalent if we consider any N contiguous integers.

Definition 3.3 (Circular convolution). *The circular convolution between two vectors $x, y \in \mathbb{C}^N$ is defined as*

$$x * y[j] := \sum_{s=0}^{N-1} x[s]\, y^{\downarrow s}[j], \quad 0 \leq j \leq N-1. \tag{30}$$

The 2D circular convolution between $X \in \mathbb{C}^{N\times N}$ and $Y \in \mathbb{C}^{N\times N}$ is defined as

$$X * Y[j_1, j_2] := \sum_{s_1=0}^{N-1} \sum_{s_2=0}^{N-1} X[s_1, s_2]\, Y^{\downarrow (s_1, s_2)}[j_1, j_2], \quad 0 \leq j_1, j_2 \leq N-1. \tag{31}$$

The following theorem is a direct consequence of the definition of convolution and Eq. (29). The 2D results follows by the same argument.

Theorem 3.4. *The action of an LTI function $\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$ on a vector $x \in \mathbb{C}^N$ is equal to the circular convolution of x and the impulse response $h_{\mathcal{F}}$ of $\mathcal{F}$,*

$$\mathcal{F}(x) = x * h_{\mathcal{F}}. \tag{32}$$

The action of a 2D LTI function $\mathcal{F} : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times N}$ on $X \in \mathbb{C}^{N\times N}$ is equal to the circular convolution of X and the impulse response $H_{\mathcal{F}}$ of $\mathcal{F}$,

$$\mathcal{F}(X) = X * H_{\mathcal{F}}. \tag{33}$$

The convolution between two vectors can be efficiently computed by multiplying their Fourier coefficients. This is illustrated in Figures 1 and 2.

Theorem 3.5 (Convolution in time is multiplication in frequency). *Let $y := x_1 * x_2$ for $x_1, x_2 \in \mathbb{C}^N$. Then the DFT of y equals*

$$\hat{y}[k] = \hat{x}_1[k]\, \hat{x}_2[k], \quad 0 \leq k \leq N-1, \tag{34}$$

$\hat{x}_1$ and $\hat{x}_2$ are the DFTs of x_1 and x_2 respectively.

Let $Y := X_1 * X_2$ for $X_1, X_2 \in \mathbb{C}^{N\times N}$. Then the 2D DFT of Y is given by

$$\widehat{Y}[k_1, k_2] = \widehat{X}_1[k_1, k_2]\, \widehat{X}_2[k_1, k_2]. \tag{35}$$

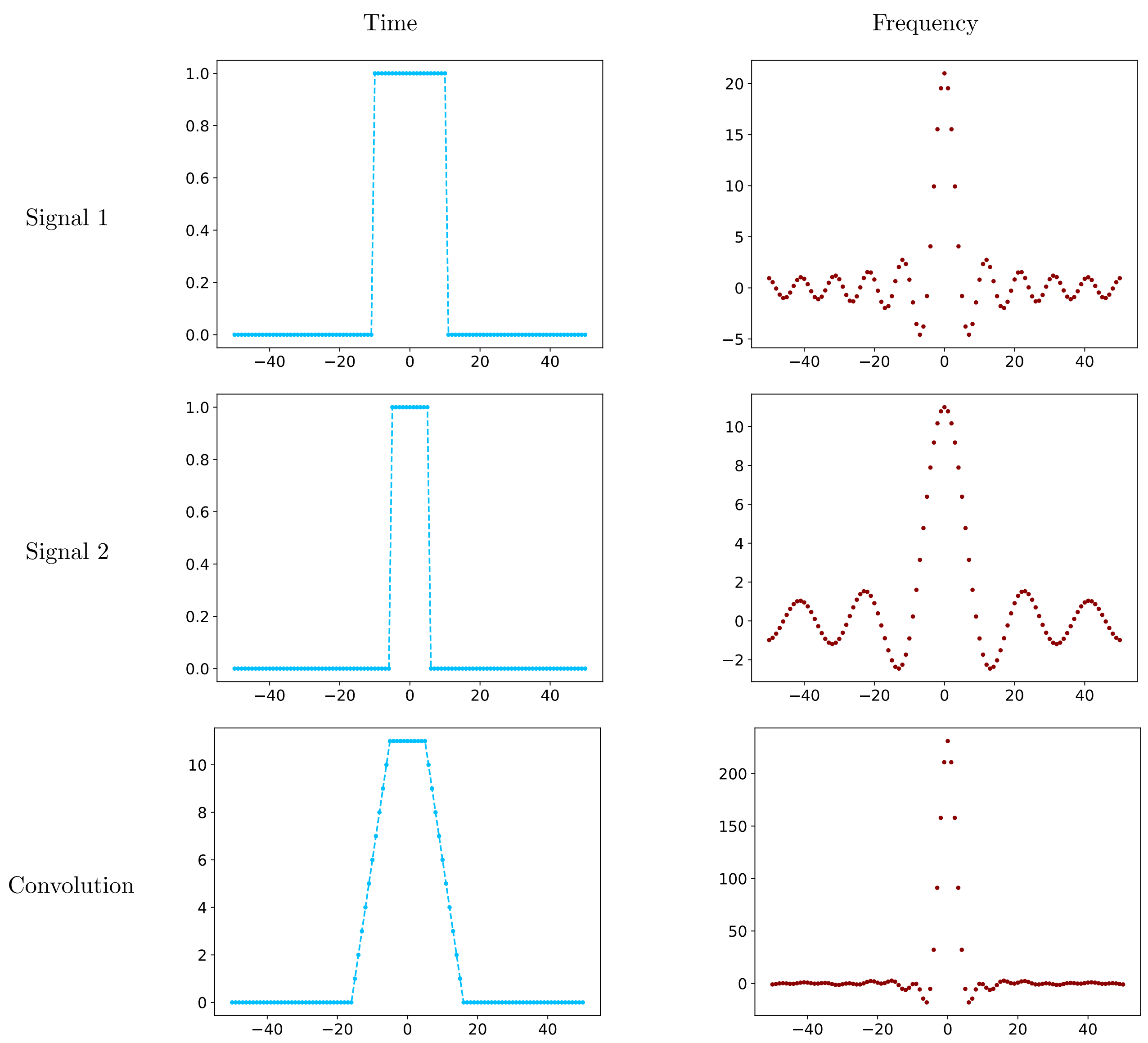


Figure 1: Convolution between two one-dimensional signals. The frequency representation of both signals and their convolution is shown on the right.

Proof. We have

$$\hat{y}[k] := \langle x_1 * x_2, \psi_k \rangle \tag{36}$$

$$= \left\langle \sum_{s=0}^{N-1} x_1[s]\, x_2^{\downarrow s}, \psi_k \right\rangle \tag{37}$$

$$= \left\langle \sum_{s=0}^{N-1} x_1[s] \frac{1}{N} \sum_{j=0}^{N-1} \exp\left(-\frac{i2\pi js}{N}\right) \hat{x}_2[j]\psi_j, \psi_k \right\rangle \quad \text{by Theorem 2.4} \tag{38}$$

$$= \sum_{j=0}^{N-1} \hat{x}_2[j] \frac{1}{N} \langle \psi_j, \psi_k \rangle \sum_{s=0}^{N-1} x_1[s] \exp\left(-\frac{i2\pi js}{N}\right) \tag{39}$$

$$= \sum_{j=0}^{N-1} \hat{x}_1[j]\hat{x}_2[j] \frac{1}{N} \langle \psi_j, \psi_k \rangle \tag{40}$$

$$= \hat{x}_1[k]\hat{x}_2[k]. \tag{41}$$

The 2D result follows from the same argument. □

A useful consequence of this theorem is that we can compute convolutions by computing the DFT of the two vectors via the FFT, multiplying their DFT coefficients, and then taking the inverse DFT again via the FFT. The complexity for N-dimensional vectors is of order $N \log N$.

The DFT of the impulse response of an LTI function is known as its transfer function. By Theorems 3.4 and 3.5 applying an LTI function to a vector just scales each of its Fourier coefficients by the corresponding entry of the transfer function.

Corollary 3.6. *Let $\hat{h}_{\mathcal{F}}$ be the DFT of the impulse response of an LTI function $\mathcal{F}$. Then for any $x \in C^N$*

$$\mathcal{F}(x) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{h}_{\mathcal{F}}[k]\hat{x}[k]\psi_k. \tag{42}$$

Let $\widehat{H}_{\mathcal{F}}$ be the 2D DFT of the impulse response of a 2D LTI function $\mathcal{F}$. Then for any $X \in C^{N\times N}$

$$\mathcal{F}(X) = \frac{1}{N^2} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \hat{H}_{\mathcal{F}}[k_1, k_2]\widehat{X}[k_1, k_2]\psi^{\text{2D}}_{k_1,k_2}. \tag{43}$$

4 Stationary signals

In this section we introduce a mathematical definition of signals with translation-invariant statistics, and study some of their properties. Signals such as sound and images often have translation-invariant structure, so this is often a useful assumption. As a motivating example, Figure 3 shows the results of applying PCA to a set of piecewise-constant signals. It turns out that the principal directions are sinusoidal! To understand why, we need to study the covariance matrices of signals with translation-invariant statistics. If we restrict our attention to second-order statistics, then such signals are called wide-sense or weak-sense stationary.

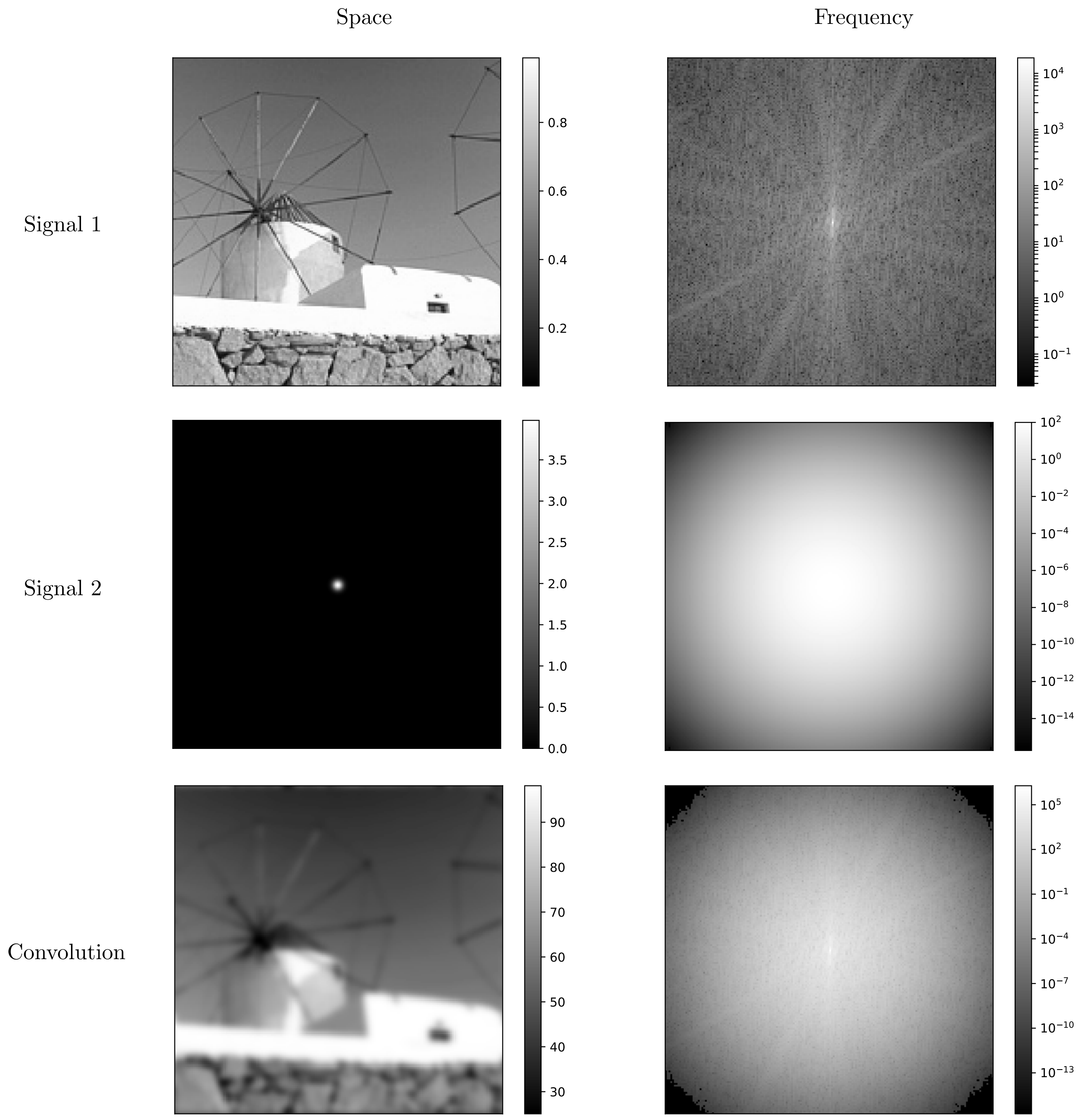


Figure 2: Image blurring caused by diffraction in optical systems is often modeled as a convolution between a high-resolution image and a blurring kernel or point-spread function. The result is a low-resolution image due to the local averaging resulting from convolving with the kernel, which in this example is Gaussian. In the frequency domain, the high-frequency components of the image are suppressed because the corresponding frequency components of the kernel are very small.

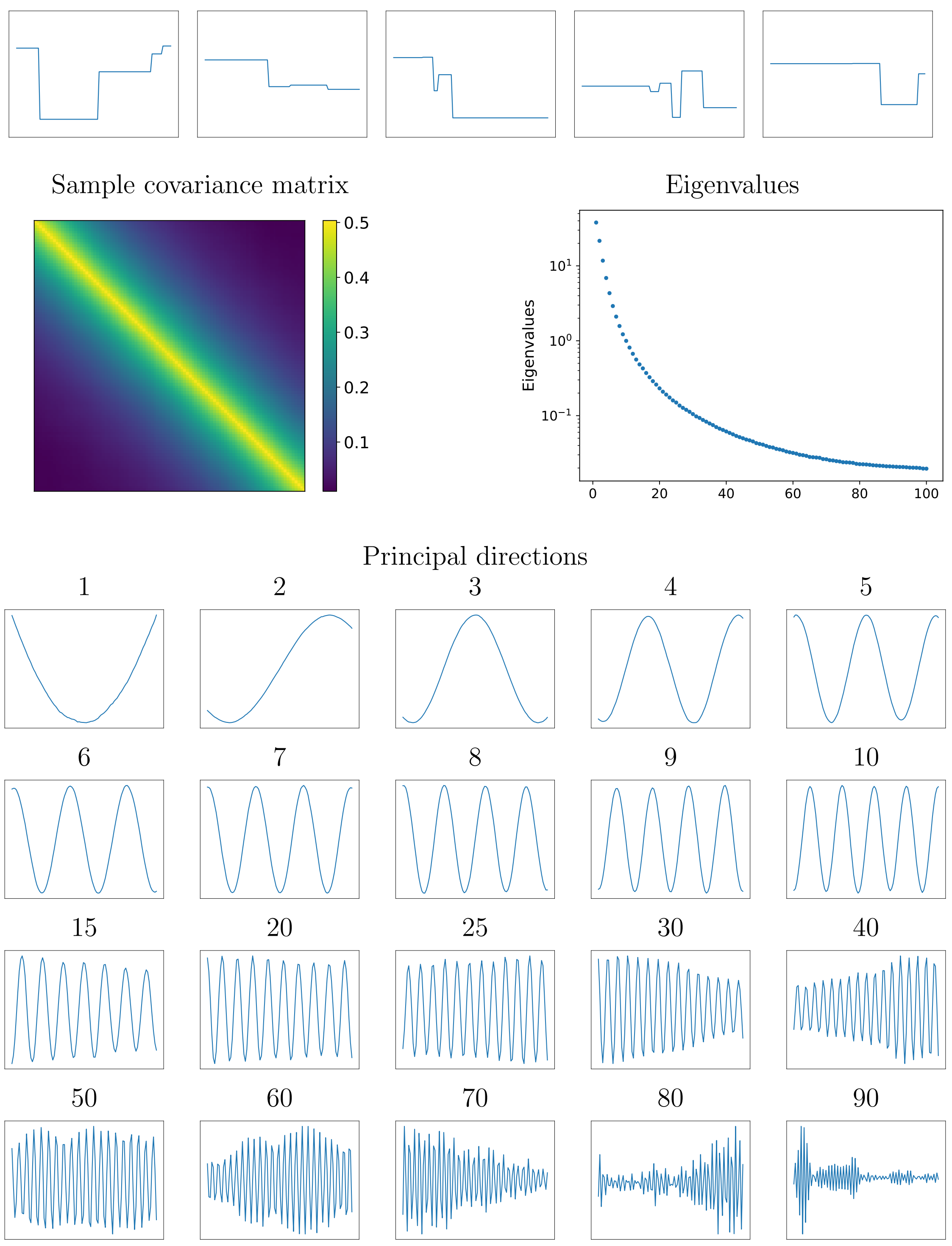


Figure 3: The top row shows examples of piecewise constant signals. The heatmap below displays the sample covariance matrix of a set of such signals. The graph to the right of the heatmap shows the eigenvalues of the sample covariance matrix. Finally, the images at the bottom show the eigenvectors, which represent the principal directions of the data.

Definition 4.1. *An N-dimensional random vector $\tilde{x}$ is wide-sense or weak-sense stationary if it satisfies two conditions. First, it has a constant mean, i.e. there exists a constant μ such that*

$$\mathrm{E}\left(\tilde{x}[j]\right) = \mu, \quad 0 \leq j \leq N-1. \tag{44}$$

Second, the covariance between any pair of entries of $\tilde{x}$ only depends on the separation between them. There exists a function $\mathrm{ac}_{\tilde{x}}$ such that

$$\mathrm{Cov}\left(\tilde{x}[j_1]\tilde{x}[j_2]\right) = \mathrm{ac}_{\tilde{x}}(j_2 - j_1 \bmod N), \quad 0 \leq j_1, j_2 \leq N-1. \tag{45}$$

The function $\mathrm{ac}_{\tilde{x}}$ is known as the autocovariance of the random vector. Note that for any integer j, $\mathrm{ac}_{\tilde{x}}(j) = \mathrm{ac}_{\tilde{x}}(-j) = \mathrm{ac}_{\tilde{x}}(N-j)$. We can write the covariance matrix of $\tilde{x}$ as

$$\Sigma_{\tilde{x}} = \begin{bmatrix} \mathrm{ac}_{\tilde{x}}(0) & \mathrm{ac}_{\tilde{x}}(N-1) & \cdots & \mathrm{ac}_{\tilde{x}}(1) \\ \mathrm{ac}_{\tilde{x}}(1) & \mathrm{ac}_{\tilde{x}}(0) & \cdots & \mathrm{ac}_{\tilde{x}}(2) \\ & & \cdots & \\ \mathrm{ac}_{\tilde{x}}(N-1) & \mathrm{ac}_{\tilde{x}}(N-2) & \cdots & \mathrm{ac}_{\tilde{x}}(0) \end{bmatrix} \tag{46}$$

$$= \begin{bmatrix} a_{\tilde{x}} & a_{\tilde{x}}^{\downarrow 1} & a_{\tilde{x}}^{\downarrow 2} & \cdots a_{\tilde{x}}^{\downarrow N-1} \end{bmatrix}, \tag{47}$$

where we define the N-dimensional autocovariance vector $a_{\tilde{x}}$ as

$$a_{\tilde{x}} := \begin{bmatrix} \mathrm{ac}_{\tilde{x}}(0) \\ \mathrm{ac}_{\tilde{x}}(1) \\ \mathrm{ac}_{\tilde{x}}(2) \\ \cdots \end{bmatrix}. \tag{48}$$

Note that we consider periodic translations (hence the mod N in Eq. 45) to simplify the analysis. If the signal is only stationary with respect to non-periodic translations, then the analysis that follows still holds approximately for most realistic situations. We refer to the review by Gray [2] for a more in-depth discussion. The covariance matrix of a wide-sense stationary signal is circulant.

Definition 4.2 (Circulant matrix)**.** *A circulant matrix is a matrix where each column vector is obtained by applying a unit circular shift to the previous column. For example,*

$$\begin{bmatrix} a & d & c & b \\ b & a & d & c \\ c & b & a & d \\ d & c & b & a \end{bmatrix} = \begin{bmatrix} h & h^{\downarrow 1} & h^{\downarrow 2} & h^{\downarrow 3} \end{bmatrix}, \tag{49}$$

where h denotes the first column, is a circulant matrix.

The following theorem establishes that the eigenvectors of circulant matrices are sinusoidal, which explains what we see in Figure 3.

Theorem 4.3. *Let $C \in \mathbb{C}^{N \times N}$ be a circulant matrix where the columns are equal to shifts of the first column $h \in \mathbb{C}^N$, then*

$$C := \frac{1}{N} F^* \,\mathrm{diag}(\hat{h}) F, \tag{50}$$

where F is the DFT matrix and the diagonal matrix $\mathrm{diag}(\hat{h})$ contains the DFT coefficients of h.

Proof. Let c be equal to the first column of the circulant matrix. By the definition of circular convolution, for any vector $x \in \mathbb{C}^N$

$$Cx = \sum_{s=0}^{N-1} x[s] \, h^{\downarrow s} \tag{51}$$

$$= h * x \tag{52}$$

$$= \frac{1}{N} F^* \operatorname{diag}(\hat{h}) F x, \tag{53}$$

where the last step follows from Theorem 3.5. This implies $C = \frac{1}{N} F^* \operatorname{diag}(\hat{h}) F$. □

In words, the eigenvalues of circulant matrices are equal to the DFT coefficients of their rows, and the eigenvectors equal complex sinusoids of corresponding frequencies. An immediate consequence is that the principal directions of wide-sense stationary signals are sinusoidal. The principal components equal the DFT coefficients of the autocovariance vector.

Corollary 4.4. *Let $\tilde{x}$ be a wide-sense stationary vector with autocovariance vector $a_{\tilde{x}}$, the eigendecomposition of the covariance matrix of $\tilde{x}$ equals*

$$\Sigma_{\tilde{x}} = \frac{1}{N} F^* \operatorname{diag}(\hat{a}_{\tilde{x}}) F, \tag{54}$$

where F is the DFT matrix, and $\hat{a}_{\tilde{x}}$ is the DFT of the autocovariance vector.

The careful reader may be surprised that the eigenvectors of the covariance matrix turn out to be complex. In fact, they are real because the autocovariance is an even function (see homework problem).

The same reasoning can be used to analyze covariance matrices of stationary 2D signals, which are vectorized. In that case, the rows of the matrix have additional structure, they can be reshaped into a 2D array that corresponds to the shape of the signals. Figure 4 shows the covariance between pixels in 32×32 images from the CIFAR-10 dataset. Pixels that are not too close to the borders of the image have associated covariances that are approximately translation-invariant and decay rather fast. Pixels close to the borders have longer ranging correlations (this is because many images have uniform backgrounds surrounding a central object). Figure 5 shows the principal directions obtained from the eigendecomposition of the covariance matrix, they are sinusoidal-looking, while displaying additional structure due to the border effects.

The fact that stationary signals have approximately circulant covariance matrices, and therefore sinusoidal principal directions, has an important consequence for dimensionality reduction: using a sinusoidal basis is close to optimal (see Section 7 in the notes on PCA). Which sinusoidal components to select depends on their associated variances (i.e. their associated eigenvalues). As shown in Figure 5 for natural images the eigenvalues decay as their associated frequency increases (see also Figure 11). Lower frequencies capture more variance than higher frequencies. Thus, up to border effects, using a basis of low-frequency sinusoids is the best we can do to perform dimensionality reduction if we care about minimizing ℓ_2-norm error. This reveals a shortcoming of optimizing this metric: fine-scale details associated to high-frequencies such as textures or sharp edges are very meaningful perceptually, but they have very small ℓ_2-norm, and are therefore

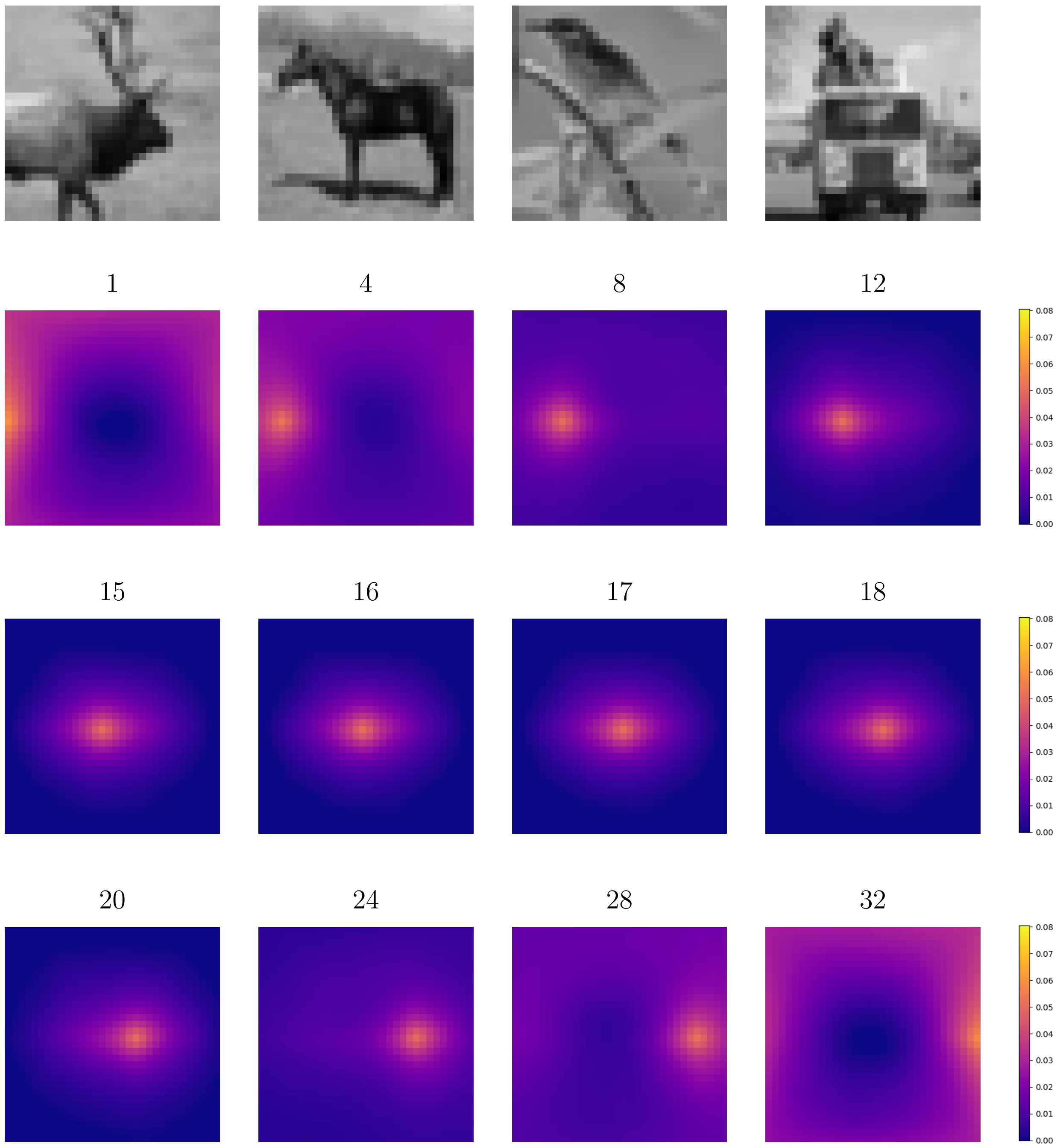


Figure 4: The top row shows examples of images from the CIFAR-10 dataset. The heatmaps below displays rows of the covariance matrix reshaped to show the values corresponding to different pixels. The pixels are located at the 16th row and at the column indicated above each heatmap.

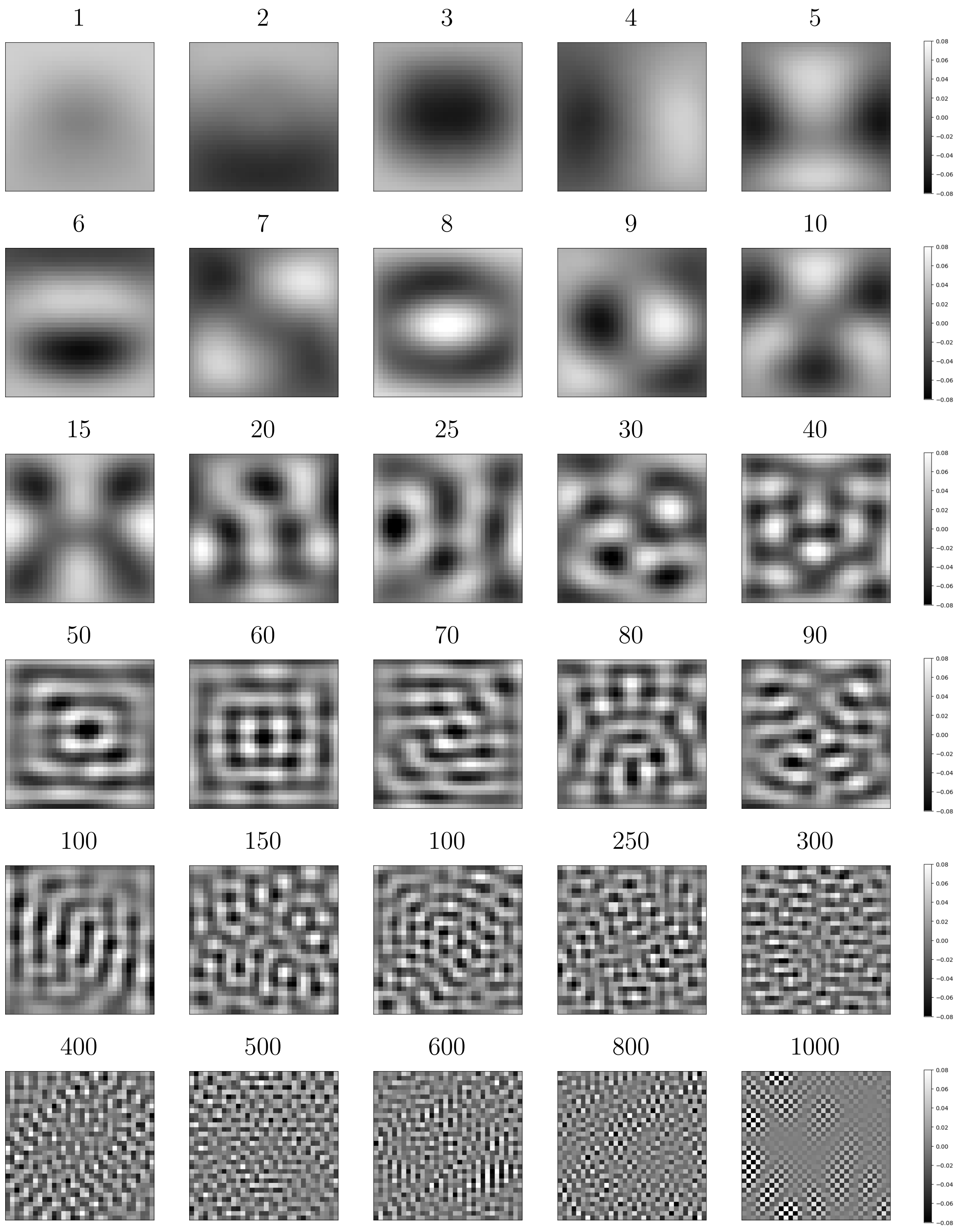


Figure 5: Eigenvectors of the sample covariance matrix of the CIFAR-10 data, reshaped to show the values corresponding to each pixel. These are the principal directions of the data.

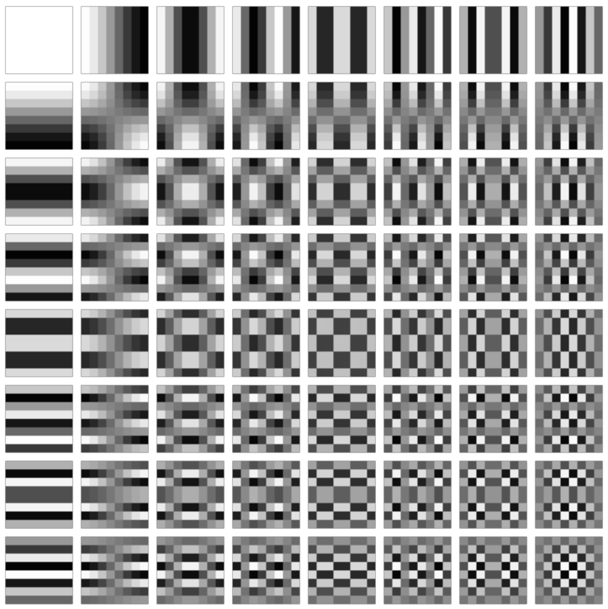


Figure 6: 8×8 DCT basis vectors.

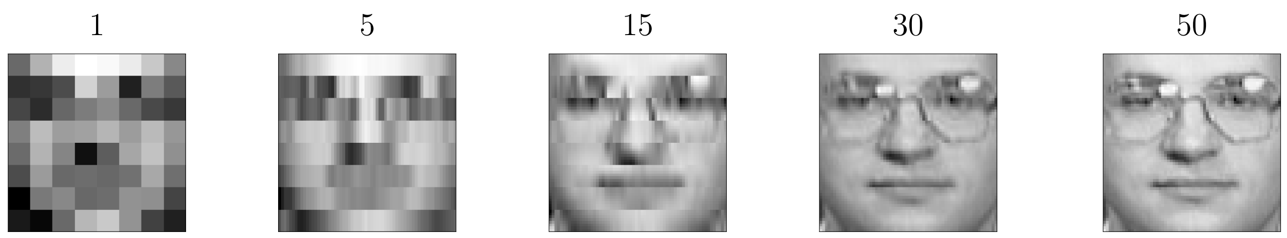


Figure 7: Result of projecting each 64-pixel patch from the image of a face onto the lowest 1, 5, 15, 30 and 50 2D DCT basis functions.

neglected. This effect is less apparent when the dimensionality reduction is applied to small patches on the image, as opposed to the whole image. This is the insight underlying the JPEG compression standard.

More specifically, JPEG uses the discrete cosine transform (DCT) to compress image patches. The DCT is a variant of the discrete Fourier transform for real signals, which assumes that the signal is symmetric (but only one half is observed). In that case the DFT components correspond to discrete cosines (the coefficients corresponding to the sines are zero due to symmetry). Figure 6 shows the basis vectors of the 8×8 2D DCT[1]. Figure 7 shows the result of dividing a natural image into 64-pixel patches and projecting it onto the span of the low-frequency DCT basis functions. Ignoring some of the high-frequency components is almost imperceptible. JPEG exploits this phenomenon to perform lossy compression of images. JPEG divides the image in 8×8 patches and then quantizes the coefficients of each DCT band differently. More bits are used for lower-frequency bands because errors in these bands are much more apparent to the human eye.

[1]The image is borrowed from `https://upload.wikimedia.org/wikipedia/commons/2/24/DCT-8x8.png`

5 Wiener filtering

In this section we consider a generalization of the regression problem, where our goal is to estimate a multidimensional signal, as opposed to just a one-dimensional response. We model the signal of interest as a random vector $\tilde{y}$, and the available measurements as a random vector $\tilde{x}$. As discussed in the notes on linear regression (Theorem 2.1), the estimate that minimizes mean-square error is the conditional mean of $\tilde{y}$ given $\tilde{x}$. However, this is usually intractable to compute unless the signals are very low dimensional, so it is necessary to constrain the regression model. Here we focus on linear models. The following theorem is the multidimensional analog to Theorem 2.3.

Theorem 5.1 (Linear minimum MSE estimation)**.** *Let $\tilde{y}$ and $\tilde{x}$ be N-dimensional zero-mean random vectors. Assume the covariance matrix of $\tilde{x}$ $\Sigma_{\tilde{x}}$ is full rank, then*

$$\Sigma_{\tilde{x}}^{-1}\Sigma_{\tilde{x}\tilde{y}} := \arg\min_{B} \mathrm{E}\left(\left|\left|\tilde{y} - B^T\tilde{x}\right|\right|_2^2\right), \tag{55}$$

where $\Sigma_{\tilde{x}\tilde{y}}$ is the cross-covariance between $\tilde{x}$ and $\tilde{y}$:

$$\Sigma_{\tilde{x}\tilde{y}} := \mathrm{E}\left(\tilde{x}\tilde{y}^T\right). \tag{56}$$

Proof. Let B_j denote the jth column of B. The cost function can be decomposed into N decoupled components

$$\mathrm{E}\left(\left|\left|\tilde{y} - B^T\tilde{x}\right|\right|_2^2\right) = \sum_{j=1}^{N} \mathrm{E}\left[\left(\tilde{y}[j] - B_j^T\tilde{x}\right)^2\right]. \tag{57}$$

By Theorem 1.4 in the notes on linear regression,

$$\Sigma_{\tilde{x}}^{-1}(\Sigma_{\tilde{x}\tilde{y}})_j = \arg\min_{B_j} \mathrm{E}\left[(\tilde{y}[j] - \tilde{x}^T B_j)^2\right], \tag{58}$$

where $(\Sigma_{\tilde{x}\tilde{y}})_j$ denotes the jth column of $\Sigma_{\tilde{x}\tilde{y}}$, which implies the result. $\square$

The problem with this estimator is that it requires building an $N \times N$ matrix, which is intractable in most real-world situations: for images N is almost always larger than 10^4, for sequences of audio of just a few seconds $N \approx 10^5$ (as we saw in the notes on the frequency domain, audio is sampled at rates of more than 40 kHz). Here we incorporate the assumption that the random vectors are stationary to obtain a more tractable model. We begin by defining joint stationarity.

Definition 5.2. *Two N-dimensional random vectors $\tilde{x}$ and $\tilde{y}$ are jointly wide-sense or weak-sense stationary if they are each wide-sense or weak-sense stationary and the cross-covariance between their entries only depends on the separation between them. There exists a function $\mathrm{cc}_{\tilde{x},\tilde{y}}$ such that*

$$\mathrm{Cov}\left(\tilde{x}[j_1]\tilde{y}[j_2]\right) = \mathrm{cc}_{\tilde{x}\tilde{y}}(j_2 - j_1 \bmod N), \quad 0 \le j_1, j_2 \le N-1. \tag{59}$$

The function $\mathrm{cc}_{\tilde{x}\tilde{y}}$ is known as the cross-covariance function of the random vectors. Note that in contrast to the covariance function, the cross-covariance function is not even. We can write the *cross-covariance matrix of $\tilde{x}$ and $\tilde{y}$ as*

$$\Sigma_{\tilde{x}\tilde{y}} = \begin{bmatrix} \mathrm{cc}_{\tilde{x}\tilde{y}}(0) & \mathrm{cc}_{\tilde{x}\tilde{y}}(N-1) & \cdots & \mathrm{cc}_{\tilde{x}\tilde{y}}(1) \\ \mathrm{cc}_{\tilde{x}\tilde{y}}(1) & \mathrm{cc}_{\tilde{x}\tilde{y}}(0) & \cdots & \mathrm{cc}_{\tilde{x}\tilde{y}}(2) \\ & & \cdots & \\ \mathrm{cc}_{\tilde{x}\tilde{y}}(N-1) & \mathrm{cc}_{\tilde{x}\tilde{y}}(N-2) & \cdots & \mathrm{cc}_{\tilde{x}\tilde{y}}(0) \end{bmatrix} \tag{60}$$

$$= \begin{bmatrix} c_{\tilde{x}\tilde{y}} & c_{\tilde{x}\tilde{y}}^{\downarrow 1} & c_{\tilde{x}\tilde{y}}^{\downarrow 2} & \cdots c_{\tilde{x}\tilde{y}}^{\downarrow N-1} \end{bmatrix}, \tag{61}$$

where we define the N-dimensional cross-covariance vector $c_{\tilde{x}}$ as

$$c_{\tilde{x}\tilde{y}} := \begin{bmatrix} \mathrm{cc}_{\tilde{x}\tilde{y}}(0) \\ \mathrm{cc}_{\tilde{x}\tilde{y}}(1) \\ \mathrm{cc}_{\tilde{x}\tilde{y}}(2) \\ \cdots \end{bmatrix}. \tag{62}$$

As in Definition 4.1 we consider periodic translations (hence the mod N in Eq. 59) to simplify the analysis. If the signal is only stationary with respect to non-periodic translations, then the analysis that follows still holds approximately for most realistic situations. The following theorem derives the best linear estimator (in terms of MSE) under a joint-stationarity assumption.

Theorem 5.3 (Wiener filter)**.** *Let $\tilde{x}$ and $\tilde{y}$ be N-dimensional zero-mean random vectors that are jointly stationary, and let $\tilde{x}_F$ and $\tilde{y}_F$ denote the DFT coefficients of $\tilde{x}$ and $\tilde{y}$ respectively. The linear estimate of $\tilde{y}$ given $\tilde{x}$ that minimizes MSE can be computed by convolving $\tilde{x}$ with the Wiener filter w, which is defined as having DFT coefficients equal to*

$$\hat{w}[k] := \frac{\mathrm{Cov}(\tilde{x}_F[k], \tilde{y}_F[k])}{\mathrm{Var}(\tilde{x}_F[k])}, \quad 0 \leq k \leq N-1, \tag{63}$$

where the Fourier coefficients can be complex so the covariance and variance are defined as

$$\mathrm{Cov}(\tilde{x}_F[k], \tilde{y}_F[k]) := \mathrm{E}\left(\tilde{x}_F[k]\overline{\tilde{y}_F[k]}\right), \tag{64}$$

$$\mathrm{Var}(\tilde{x}_F[k]) := \mathrm{E}\left(|\tilde{x}_F[k]|^2\right), \quad 0 \leq k \leq N-1. \tag{65}$$

Proof. By Corollary 4.4

$$\Sigma_{\tilde{x}} = \frac{1}{N} F^* \operatorname{diag}(\hat{a}_{\tilde{x}}) F, \tag{66}$$

and by Theorem 4.3

$$\Sigma_{\tilde{x}\tilde{y}} = \frac{1}{N} F^* \operatorname{diag}(\hat{c}_{\tilde{x}}) F, \tag{67}$$

because $\Sigma_{\tilde{x}\tilde{y}}$ is a circulant matrix. By Theorem 5.1 the optimal linear transformation is given by

$$\Sigma_{\tilde{x}\tilde{y}}^T \Sigma_{\tilde{x}}^{-1} = \left(\frac{1}{N} F^* \operatorname{diag}(\hat{a}_{\tilde{x}}) F\right)^{-1} \frac{1}{N} F^* \operatorname{diag}(\hat{c}_{\tilde{x}}) F \tag{68}$$

$$= \frac{1}{N} F^* \operatorname{diag}(\hat{a}_{\tilde{x}}^{-1}) F \frac{1}{N} F^* \operatorname{diag}(\hat{c}_{\tilde{x}}) F \tag{69}$$

$$= \frac{1}{N} F^* \operatorname{diag}(\hat{a}_{\tilde{x}}^{-1} \hat{c}_{\tilde{x}}) F. \tag{70}$$

Let us consider the covariance matrix of the DFT coefficients of $\tilde{x}$. By linearity of expectation and (66)

$$\begin{aligned}
\Sigma_{\tilde{x}_F} &:= \mathrm{E}\left(F\tilde{x}(F\tilde{x})^*\right) && (71)\\
&= F\mathrm{E}\left(\tilde{x}\tilde{x}^T\right)F^* && (72)\\
&= F\Sigma_{\tilde{x}}F^* && (73)\\
&= F\frac{1}{N}F^*\operatorname{diag}(\hat{a}_{\tilde{x}})FF^* && (74)\\
&= N\operatorname{diag}(\hat{a}_{\tilde{x}}), && (75)
\end{aligned}$$

so the Fourier coefficients are uncorrelated and

$$\hat{a}_{\tilde{x}}[k] = \frac{\operatorname{Var}(\tilde{x}_F[k])}{N}, \quad 0 \leq k \leq N-1. \tag{76}$$

Similarly, by linearity of expectation and (67)

$$\begin{aligned}
\Sigma_{\tilde{x}_F\tilde{y}_F} &:= \mathrm{E}\left(F\tilde{x}(F\tilde{y})^*\right) && (77)\\
&= F\Sigma_{\tilde{x}\tilde{y}}F^* && (78)\\
&= N\operatorname{diag}(\hat{c}_{\tilde{x}\tilde{y}}), && (79)
\end{aligned}$$

so

$$\hat{c}_{\tilde{x}\tilde{y}}[k] = \frac{\operatorname{Cov}(\tilde{x}_F[k], \tilde{y}_F[k])}{N}, \quad 0 \leq k \leq N-1. \tag{80}$$

Plugging Eqs. (76) and (80) into Eq. (70) completes the proof. □

In words, under a joint-stationarity assumption, the linear minimum MSE estimate can be achieved by computing the linear minimum MSE estimate of each DFT coefficient of the signal based only on the corresponding DFT coefficient of the data. The reason is that the different Fourier coefficients are all uncorrelated; the covariance and cross-covariance matrix of the DFT coefficients are diagonal.

In practice, we use the sample variance and covariance of the DFT coefficients to construct the Wiener filter. Interestingly, one can arrive at exactly the same estimator from a data-driven perspective, analogously to the relation between linear MMSE estimation and least squares. Assume that we have available n pairs of observed signals: (y_1, x_1), (y_2, x_2), ..., (y_n, x_n), where $y_i \in \mathbb{R}^N$ and $x_i \in \mathbb{R}^N$ for $1 \leq i \leq n$. If we have reason to believe that the signals have translation-invariant statistics, a reasonable way to build an estimator is to compute the LTI model that minimizes the least-squares cost function on the training set. It turns out that this is just the Wiener filter.

Theorem 5.4 (Data-driven Wiener filter)**.** *Let (x_1, y_1), ..., (x_n, y_n) be a training set of n pairs of observed signals of dimension N, which are all centered to have zero sample mean. The solution to the ℓ_2-norm loss optimization problem*

$$w := \arg\min_{v \in \mathbb{C}^N} \sum_{j=1}^{n} ||y_j - v * x_j||_2^2 \tag{81}$$

is the Wiener filter w with transfer function given by

$$\hat{w} = \frac{\operatorname{cov}\left(\hat{\mathcal{X}}[k], \hat{\mathcal{Y}}[k]\right)}{\operatorname{var}\left(\hat{\mathcal{X}}[k]\right)}, \quad 0 \leq k \leq N-1, \tag{82}$$

where

$$\operatorname{cov}\left(\hat{\mathcal{X}}[k], \hat{\mathcal{Y}}[k]\right) := \frac{1}{n}\sum_{j=1}^{n} \hat{x}_j[k]\overline{\hat{y}_j[k]}, \tag{83}$$

$$\operatorname{var}\left(\hat{\mathcal{X}}[k]\right) := \frac{1}{n}\sum_{j=1}^{n} \left|\hat{x}_j[k]\right|^2, \quad 0 \leq k \leq N-1. \tag{84}$$

Proof. The cost function can be rewritten in terms of the Fourier coefficients of the signals in the training set by Theorem 3.5,

$$\sum_{j=1}^{n} ||y_j - v * x_j||_2^2 = \sum_{j=1}^{n} \left|\left|\frac{1}{\sqrt{N}} F^*(\hat{y}_j - \hat{v} \circ \hat{x}_j)\right|\right|_2^2 \tag{85}$$

$$= \frac{1}{N^2}\sum_{j=1}^{n} ||\hat{y}_j - \hat{v} \circ \hat{x}_j||_2^2 \tag{86}$$

$$= \frac{1}{N^2}\sum_{j=1}^{n}\sum_{k=1}^{N} \left|\hat{y}_j[k] - \hat{v}[k]\hat{x}_j[k]\right|^2 := \frac{1}{N^2}\sum_{k=1}^{N} C_k\left(\hat{v}[k]\right). \tag{87}$$

This decouples the cost function into N terms, each of which only depends on one Fourier coefficient.

$$C_k(\alpha) := \frac{1}{2}\sum_{j=1}^{n} \left|\hat{y}_j[k] - \alpha\hat{x}_j[k]\right|^2 \tag{88}$$

$$= \sum_{j=1}^{n} \left|\hat{y}_j[k]\right|^2 - 2\operatorname{Re}\left\{\hat{y}_j[k]\overline{\hat{x}_j[k]}\right\}\alpha_R - 2\operatorname{Im}\left\{\hat{y}_j[k]\overline{\hat{x}_j[k]}\right\}\alpha_I + \left|\hat{x}_j[k]\right|^2\left(\alpha_R^2 + \alpha_I^2\right),$$

where α_R and α_I denote the real and imaginary part of α respectively. Setting the derivatives with respect to these parameters to zero, we obtain

$$\arg\min_{\alpha \in \mathbb{C}} C_k(\alpha) = \frac{\sum_{j=1}^{n} \operatorname{Re}\left\{\hat{x}_j[k]\overline{\hat{y}_j[k]}\right\}}{\sum_{j=1}^{n} \left|\hat{y}_j[k]\right|^2} + i\frac{\sum_{j=1}^{n} \operatorname{Im}\left\{\hat{x}_j[k]\overline{\hat{y}_j[k]}\right\}}{\sum_{j=1}^{n} \left|\hat{y}_j[k]\right|^2}. \tag{89}$$

□

Optimal linear translation-invariant filtering is equivalent to scaling each Fourier coefficient of the noisy signal by a fixed constant. In the case of additive iid Gaussian noise, the scaling has a very simple expression.

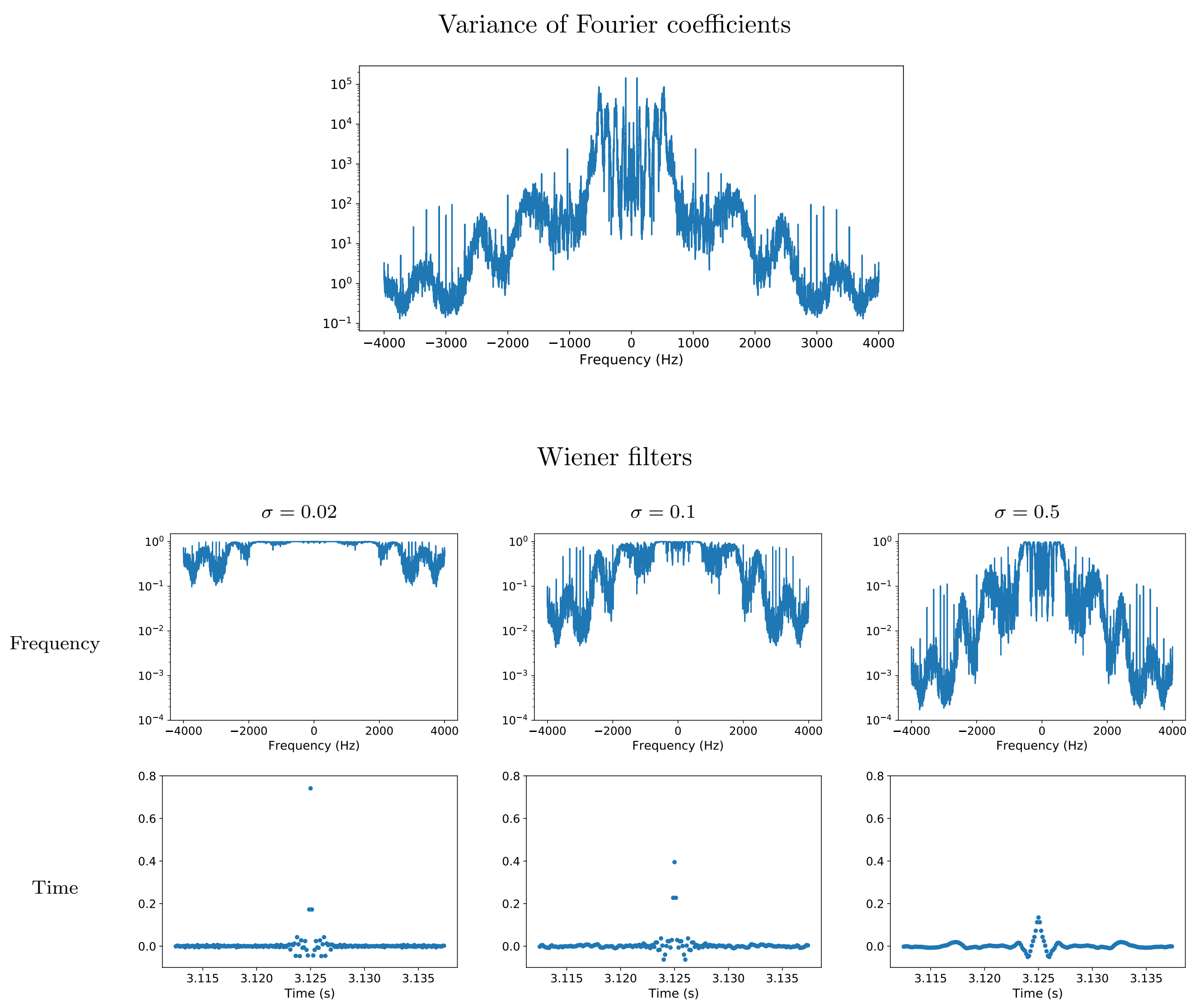


Figure 8: The plot at the top shows the variance of the Fourier coefficients of a set of audio signals, in which people say *yes* and *no* alternatively in Hebrew. The corresponding Wiener filters for different noise levels are shown below in the frequency and time domains.

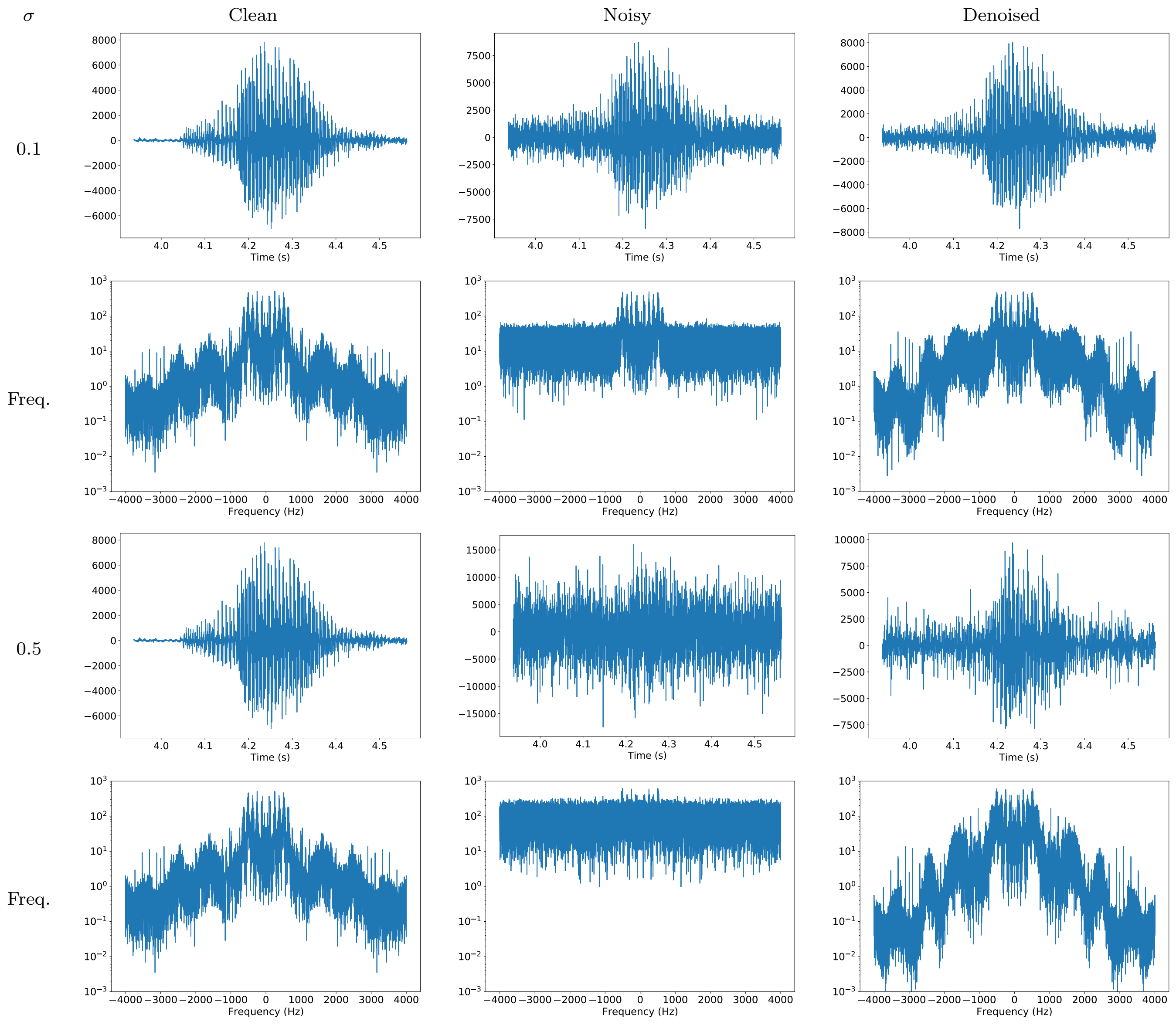


Figure 9: Result of applying the filters in Figure 8 to two test audio signals with different noise levels. Click on these links to listen to the audio: clean signal, noisy ($\sigma = 0.1$), denoised ($\sigma = 0.1$), noisy ($\sigma = 0.5$), denoised ($\sigma = 0.5$).

$\sigma = 0.1$

$\sigma = 0.5$

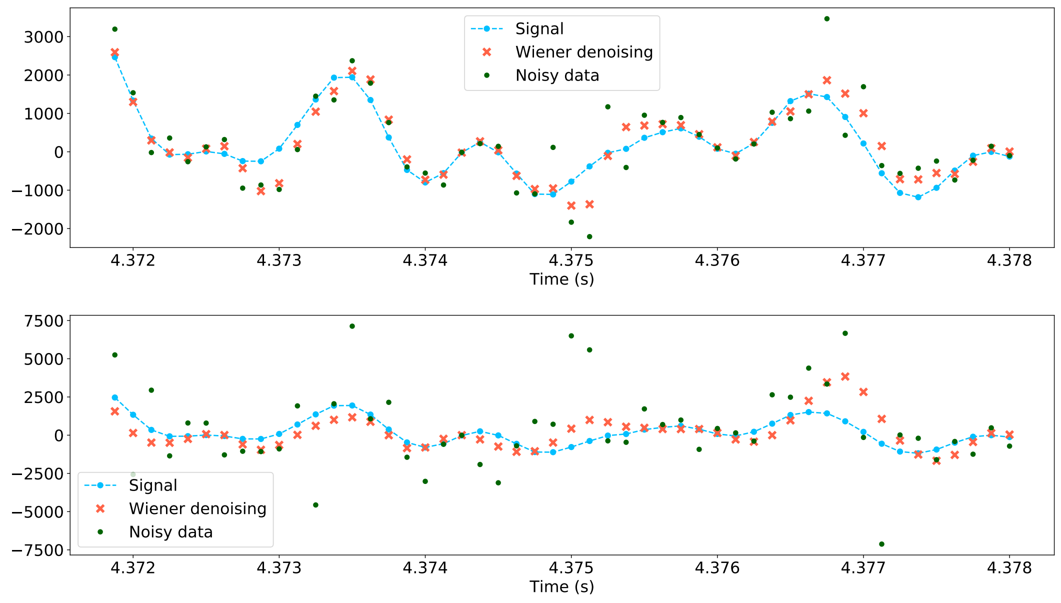


Figure 10: Zoomed plots of the denoising results shown in Figure 9.

Example 5.5 (Denoising via Wiener filtering)**.** Let us consider a denoising problem where the data are modeled probabilistically as n samples from a random vector

$$\tilde{x} = \tilde{y} + \tilde{z}, \tag{90}$$

where $\tilde{z}$ is zero-mean Gaussian noise with variance σ^2, which is independent of the signal $\tilde{y}$. The goal is to estimate the signal from the noisy measurements.

Theorem 8.6 in the notes on PCA can be extended to complex random vectors. This implies that the DFT of $\tilde{z}$ is a Gaussian vector with covariance matrix $F\sigma^2 I F^* = N\sigma^2 I$ and mean zero. The Fourier coefficients of the noise, which we denote as $\tilde{z}_F$, are consequently iid complex Gaussian random variables with zero mean and variance $N\sigma^2$. This implies

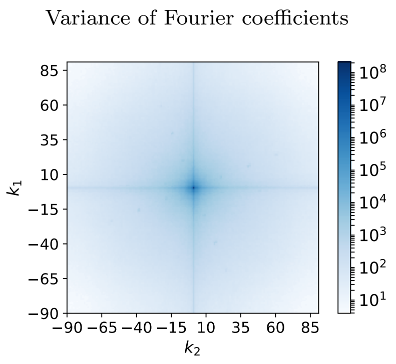
$$\begin{align} \operatorname{Cov}(\tilde{x}_F[k], \tilde{y}_F[k]) &= \operatorname{E}\left(\tilde{x}_F[k]\overline{\tilde{y}_F[k]}\right) \tag{91}\\ &= \operatorname{E}\left(\tilde{y}_F[k]\overline{\tilde{y}_F[k]}\right) + \operatorname{E}\left(\tilde{z}_F[k]\overline{\tilde{y}_F[k]}\right) \tag{92}\\ &= \operatorname{Var}\left(\tilde{y}_F[k]\right), \tag{93} \end{align}$$

and

$$\begin{align} \operatorname{Var}(\tilde{x}_F[k]) &= \operatorname{Var}\left(\tilde{y}_F[k]\right) + \operatorname{Var}\left(\tilde{z}_F[k]\right) \tag{94}\\ &= \operatorname{Var}\left(\tilde{y}_F[k]\right) + N\sigma^2. \tag{95} \end{align}$$

By Theorem 5.3, the Fourier coefficients of the optimal denoising Wiener filter are given by

$$\hat{w}[k] = \frac{\operatorname{Var}\left(\tilde{y}_F[k]\right)}{\operatorname{Var}\left(\tilde{y}_F[k]\right) + N\sigma^2}, \quad 0 \leq k \leq N-1. \tag{96}$$



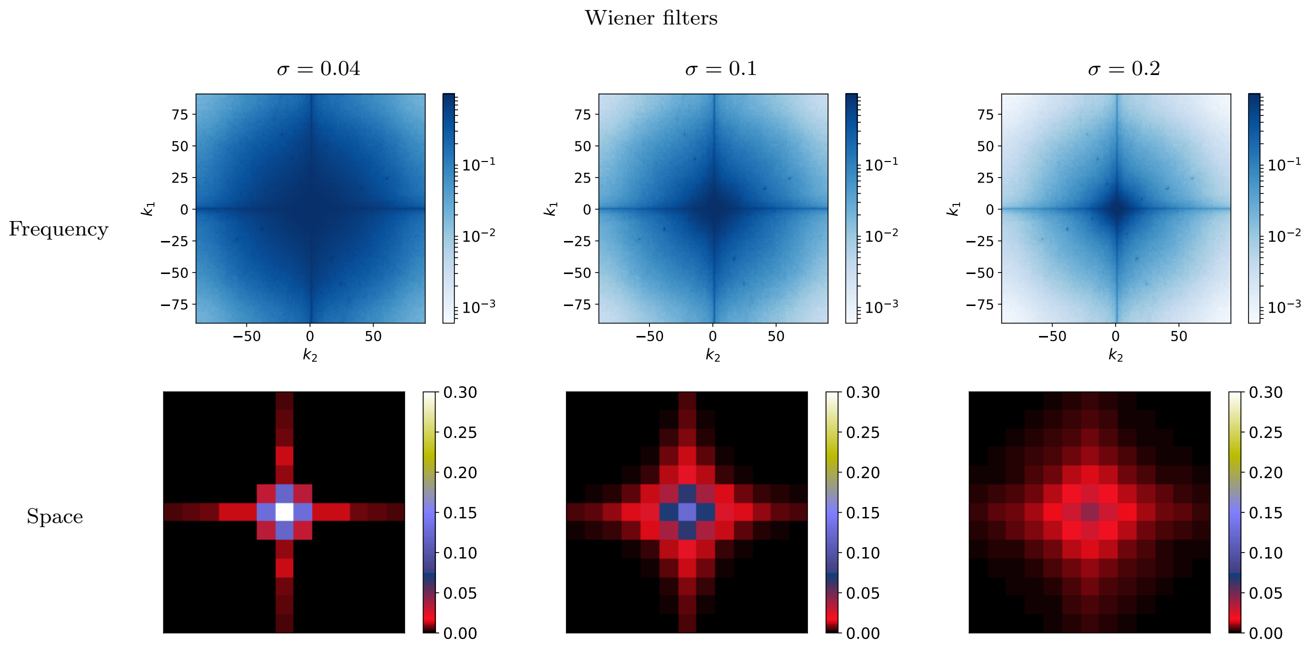


Figure 11: The heatmap at the top shows the variance of the Fourier coefficients of a set of natural images. The corresponding Wiener filters for different noise levels are shown below in the frequency and spatial domains.

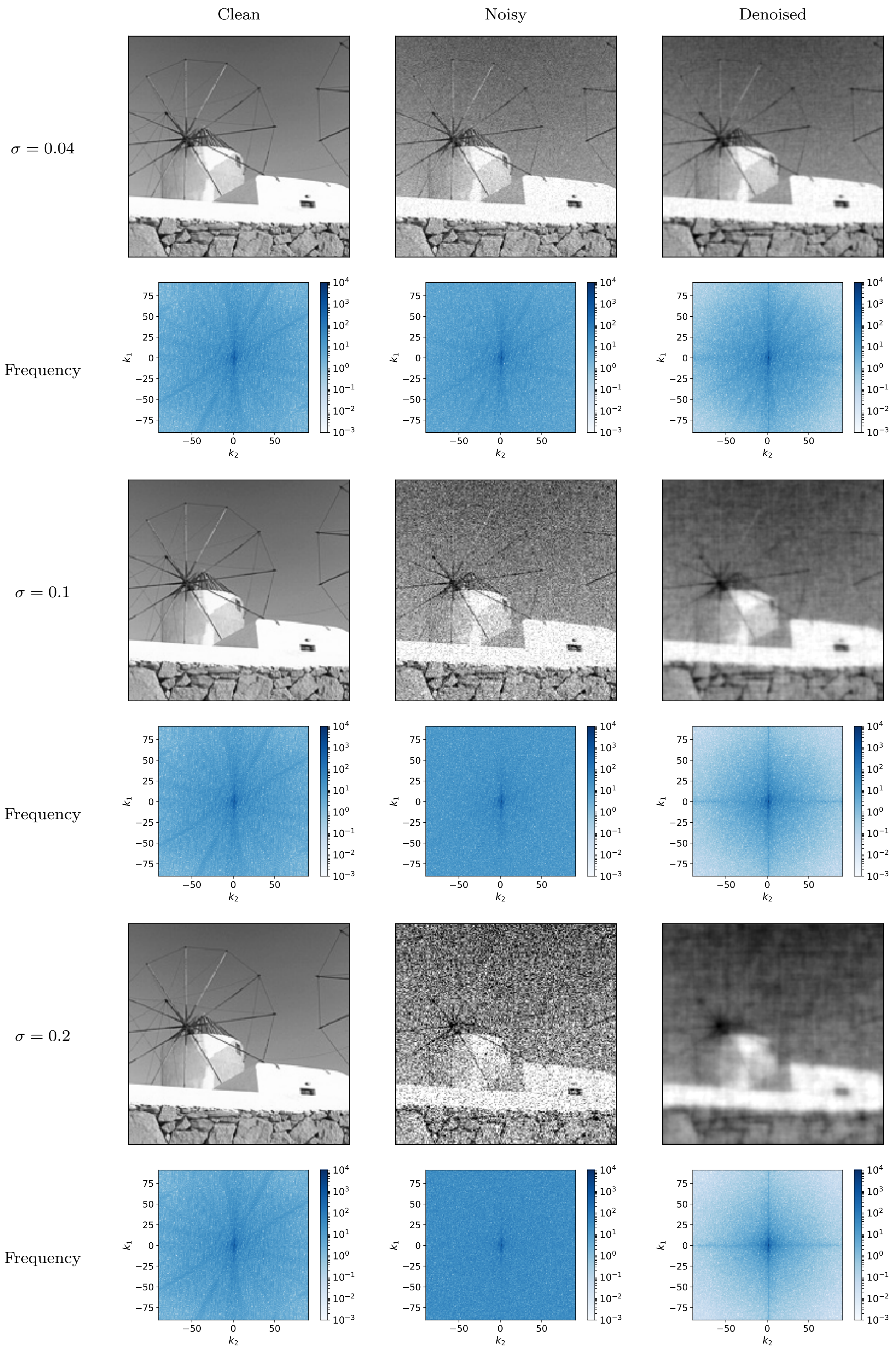


Figure 12: Result of applying the filters in Figure 11 to a test image with different noise levels.

Figures 8 and 11 show the variance of the Fourier coefficients of a set of signals, and the corresponding Wiener filters for different noise levels. When the noise level is low, the filter is concentrated at the origin, which means that each entry of the estimated signal is almost equal to the noisy measurement. As the noise rises, the filter suppresses the frequencies at which the signals in the training set have less energy. For audio and natural images, this usually results in filtering out high frequencies. In the time domain, this results in a filter that averages over more neighboring entries to produce the estimate. This is apparent in the denoising examples shown in Figures 9, 10 and 12. △

6 Convolutional neural networks for image denoising

In the lecture notes on Fourier we described how to denoise signals using a linear translation-invariant filter. In the lecture notes on signal representations we showed that thresholding-based techniques produce better results. These techniques apply a linear transformation to the signal, which yields a sparse representation that is thresholded to remove the noise. In this section we describe an approach to learn more general nonlinear denoisers based on deep learning.

Convolutional neural networks (CNNs) have been extremely successful in image classification [3]. They are translation-invariant nonlinear models implemented by a series of convolutions with different filters interleaved by pointwise nonlinearities. This yields a model with a large number of parameters, which are learned by minimizing a cost function related to the task of interest. In the case of denoising, the cost function is of the form

$$\min_{W_1,\ldots,W_L} \sum_{j=1}^{n} \left|\left| y^{[j]} - W_L r\left(W_{L-1} \cdots r\left(W_2 r\left(W_1 x^{[j]}\right)\right)\right)\right|\right|_2^2, \tag{97}$$

where $(y^{[1]}, x^{[1]})$, ..., $(y^{[n]}, x^{[n]})$ is a training set of n pairs of clean and noisy signals of dimension N. The nonlinearity r is chosen to be a pointwise rectifier linear unit, for any $\vec{v} \in \mathbb{R}^N$ $r(\vec{v})[i] := \max\{0, \vec{v}[i]\}$ for $1 \leq i \leq N$. The matrices W_1, ..., W_L contain multiple different convolutional filters each, which are applied to the output of the previous layer after rectification. CNNs can also have additive parameters, but this is often unnecessary for image denoising (see [4]). The number of parameters in these models is very large (around half a million), and therefore requires a large training set of image patches (note however that these can be extracted from a set of just a few hundred clean images). To make this tractable, stochastic gradient descent (see the notes on convex optimization) is applied to solve the minimization problem. For more details we refer to [5] (see also [1] for an excellent introduction to deep learning).

The results of deep-learning based denoising are compared in Figure 13 to the Wiener filtering, and block-thresholding wavelet coefficients. The CNN produces significantly better results, removing more noise while preserving image structure very effectively. In order to understand more intuitively what the DNCNN is doing, we can consider the Jacobian matrix of the function

$$f(x) \approx W_L r\left(W_{L-1} r\left(\ldots W_2 r\left(W_1 x\right)\right)\right) \tag{98}$$

for a fixed image y. The Jacobian matrix provides a linear approximation to the function. The rows of this matrix can be interpreted as denoising filters adapted to the specific structure of the

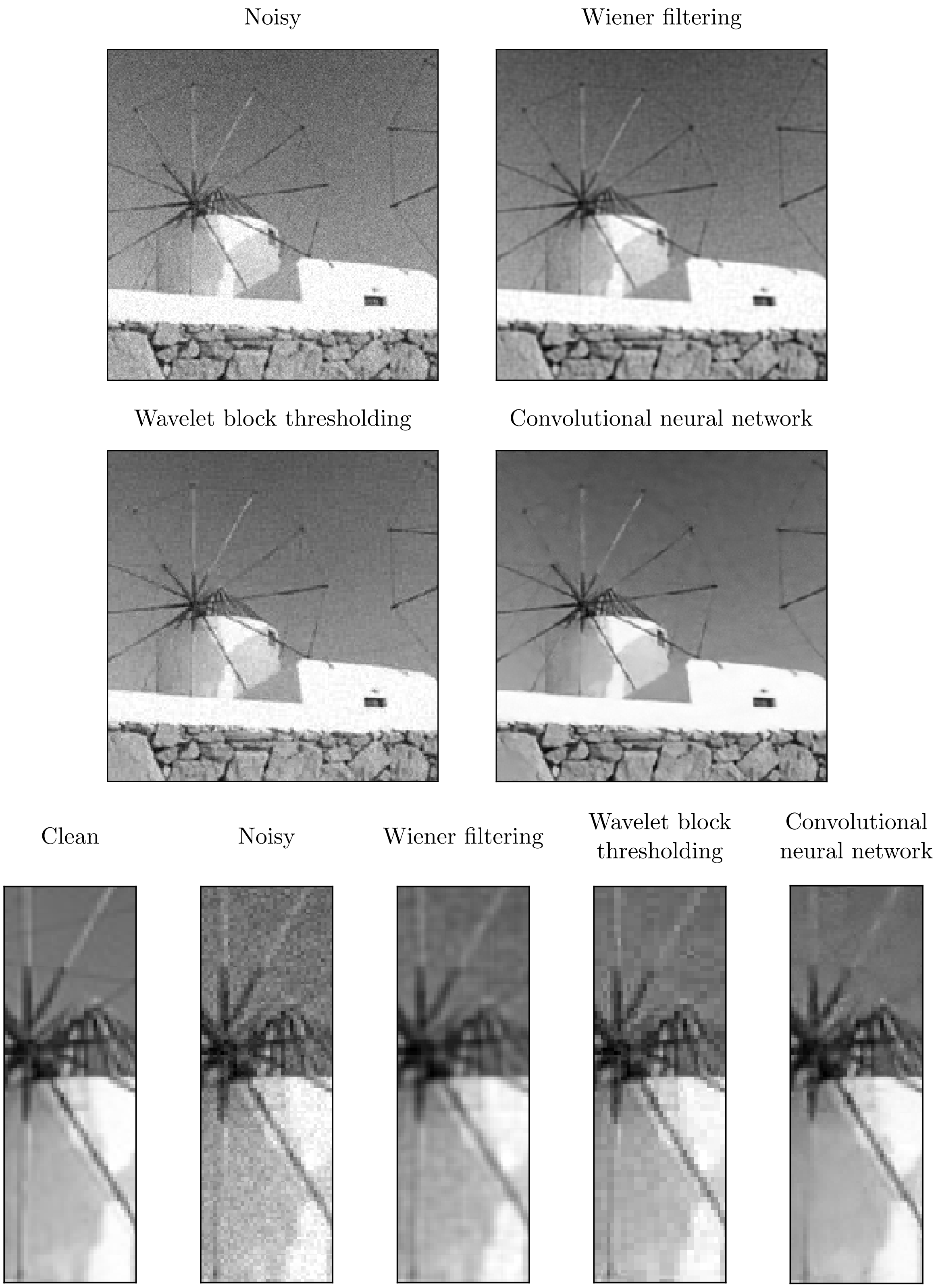

Noisy
Wiener filtering
Wavelet block thresholding
Convolutional neural network
Clean
Noisy
Wiener filtering
Wavelet block thresholding
Convolutional neural network

Figure 13: Denoising results for Wiener filtering, wavelet block thresholding and a convolutional neural network for iid Gaussian noise with standard deviation $\sigma := 0.04$.

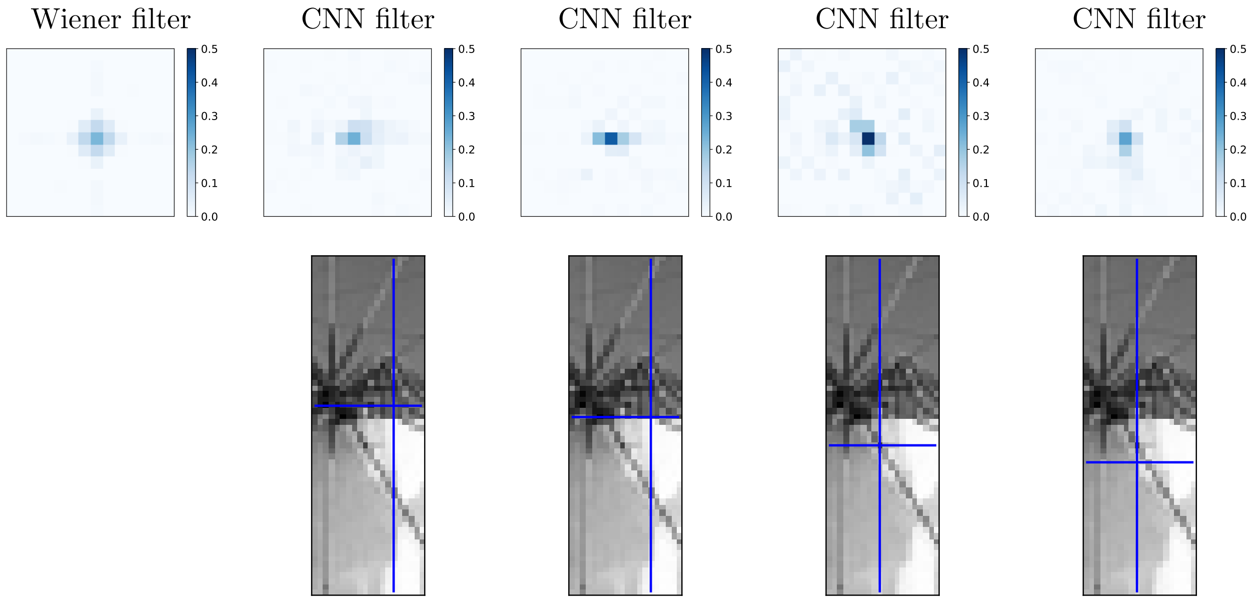


Figure 14: The top row shows a Wiener filter together with rows of the Jacobian of a CNN, which can be interpreted as nonlinear filters tied to the particular image. The second row shows the location of the pixels corresponding to each of the rows of the Jacobian.

image (see [4] for more details on this visualization technique). Figure 14 shows some of these filters for different locations in an image. The filters adapt to edges and other features.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[2] R. M. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, 2(3):155–239, 2006.

[3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli, and C. Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *International Conference on Learning Representations*, 2019.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.