# CNNs from the perspective of signal processing

Let us now see how CNNs arise naturally as a powerful way of representing signals. A lot of the mathematical ideas for this section come from:

(i) Mallat - "Group Invariant Scattering" (2012)

(ii) Bruna & Mallat - "Invariant Scattering Convolution Network" (2013)

(iii) ... and several subsequent papers

Per our previous discussions, suppose we are looking for a representation $\Phi_J(x)$ of signal type data, which we model as $x : \mathbb{R} \to \mathbb{R}$. Define

$$\|x\|_2 = \int_{\mathbb{R}} |x(u)|^2 \, du < +\infty$$

We want $\Phi_J(x)$ to have the following properties:

(a) Translation invariance up to the scale $2^J$

(b) Stability to diffeomorphisms

Combining (a) and (b) and recalling that for $\tau \in C^2(\mathbb{R})$ w/ $\|\tau'\|_\infty \leq \frac{1}{2}$ we defined

$$x_\tau(u) = x(u - \tau(u))$$

we want:

$$\|\Phi(x) - \Phi(x_\tau)\|_2 \leq C \cdot \left[2^{-J}\|\tau\|_\infty + \|\tau'\|_\infty + \|\tau''\|_\infty\right] \|x\|_2$$

But is this enough? Consider the representation:

$$\Phi(x) = \int_{\mathbb{R}} x(u) \, du$$

CoV: $v = u - t$

We have:

$$\Phi(x_t) = \int_{\mathbb{R}} x_t(u) \, du = \int_{\mathbb{R}} x(u-t) \, du = \int_{\mathbb{R}} x(v) \, dv$$

$\Rightarrow \Phi(x_t) = \Phi(x)$ and so $\Phi(x)$ is translation invariant

We also have:

$$\Phi(x_\tau) = \int_{\mathbb{R}} x_\tau(u) \, du = \int_{\mathbb{R}} x(u - \tau(u)) \, du$$

$v = u - \tau(u)$

$dv = [1 - \tau'(u)] \, du$

$$= \int_{\mathbb{R}} \frac{x(v)}{1 - \tau'(u)} \, dv$$

(u depends on v)

Therefore:

$$\Phi(x) - \Phi(x_\tau) = \int_{\mathbb{R}} x(v) \, dv - \int_{\mathbb{R}} \frac{x(v)}{1 - \tau'(u)} \, dv$$

$$= \int_{\mathbb{R}} \left[1 - \frac{1}{1 - \tau'(u)}\right] x(v) \, dv = \int_{\mathbb{R}} \frac{-\tau'(u)}{1 - \tau'(u)} \cdot x(v) \, dv$$

$$\Rightarrow |\Phi(x) - \Phi(x_\tau)| = \left| \int_{\mathbb{R}} -\frac{\tau'(u)}{1-\tau'(u)} x(v)\,dv \right| \leq \int_{\mathbb{R}} \left| \frac{\tau'(u)}{1-\tau'(u)} \right| |x(v)|\,dv$$

$$\leq 2\|\tau'\|_\infty \int_{\mathbb{R}} |x(v)|\,dv = 2\|\tau'\|_\infty \|x\|_1 \qquad (*)$$

Therefor $\Phi(x)$ is translation invariant and stable to diffeomorphisms as encoded by $(*)$. But $\Phi(x)$ is not a very good reprentation because it is just the integral of $x$. Many different signals have the same integral. Therefore to (a) and (b) we must add another condition:

(c) The representation retains enough information in $x$ to perform the task.

Condition (c) is not as precise as (a) and (b). A precise and very strong version of (c) is:

$$\Phi(x) = \Phi(y) \iff y = x_t \text{ for some } t \qquad (**)$$

Equation $(**)$ says $\Phi(x)$ is invertible up to translations. While this is mathematically precise, it may also take things too far. Indeed, in many classification tasks, $\Phi(x)$ being invertible is not a requirement for good classification results. We will instead be content to develop a systematic way of adding new information into $\Phi(x)$ while maintaining properties (a) (translation invariance) and (b) (stability to diffeomorphisms).

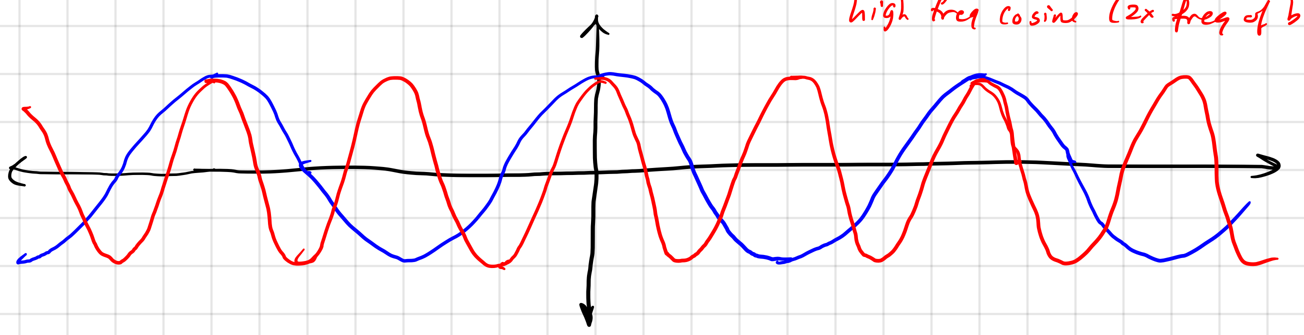
A key to understanding local translation invariance and diffeomorphism stability is through frequency representations of signals $x: \mathbb{R} \to \mathbb{R}$. For example, in a piece of music, we listen to the piece in time, but another way of representing the piece is through the notes, or frequencies, contained in it. The Fourier transform is the mathematically precise way to do this. Define a complex valued sinusoid at the frequency $\omega$ as:

$$e_\omega(u) = e^{i\omega u} = \cos(\omega u) + i\sin(\omega u), \qquad i = \sqrt{-1}$$

The frequency is $\omega$ because the cosine and sine functions are periodic with period $2\pi/\omega$. Thus the higher $\omega$, the faster the cosine and sine waves oscillate.

low freq cosine

high freq cosine (2x freq of blue)



The Fourier transform of $x : \mathbb{R} \to \mathbb{R}$ w/ $\int_{\mathbb{R}} |x(u)|\, du < \infty$ computes:

$$\hat{x}(\omega) = \langle x, e_\omega \rangle = \int_{\mathbb{R}} x(u)\, e^{-i\omega u}\, du \quad , \quad \omega \in \mathbb{R}$$

It thus tests the signal $x$ against each sinusoid, and records which frequencies are present in $x$ through $\hat{x}$.

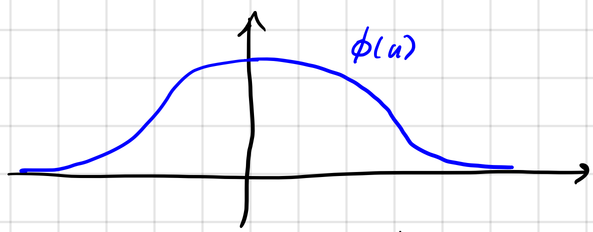
Assume $\int_{\mathbb{R}} |\hat{x}(\omega)|\, d\omega < \infty$. Then knowing $\hat{x}$ is equivalent to knowing $x$ since:

$$x(u) = \int_{\mathbb{R}} \hat{x}(\omega)\, e^{i\omega u}\, d\omega$$

We will let $\phi : \mathbb{R} \to \mathbb{R}$ denote a low pass filter. This means:

$$\hat{\phi}(\omega) = 0 \text{ for all } |\omega| > \pi \quad \text{and} \quad 1 = \hat{\phi}(0) \geq |\hat{\phi}(\omega)|$$

Intuitively, $\phi$ will be a "bump function":



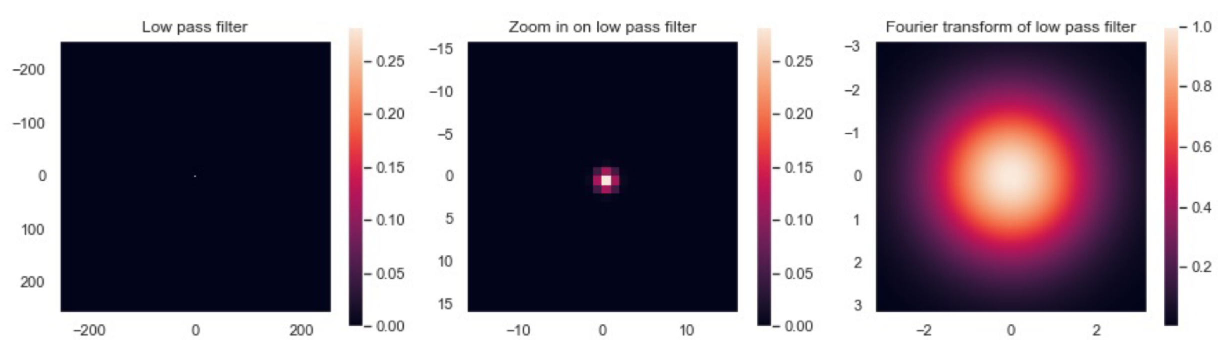

Filtering $x$ with $\phi$ computes: $x * \phi$

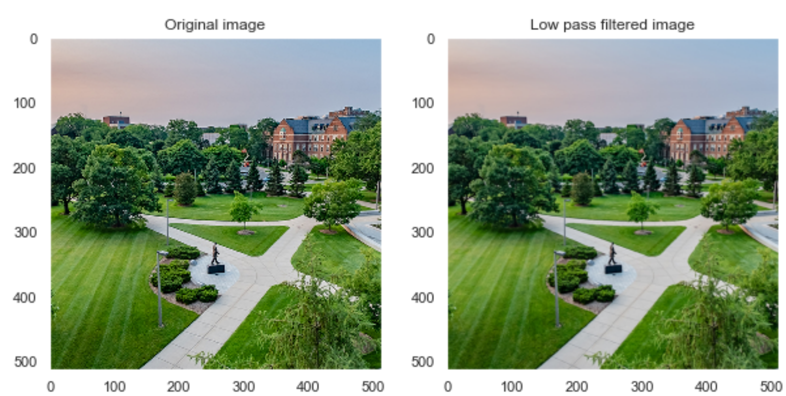
The resulting signal $x * \phi$ is a smoothed, or blurred, version of $x$. Since

$$\widehat{(x * \phi)}(\omega) = \hat{x}(\omega)\, \hat{\phi}(\omega) \qquad \text{(Fourier convolution theorem)}$$

It keeps only the low frequencies of $x$ contained in $[-\pi, \pi]$.

Here is an example:



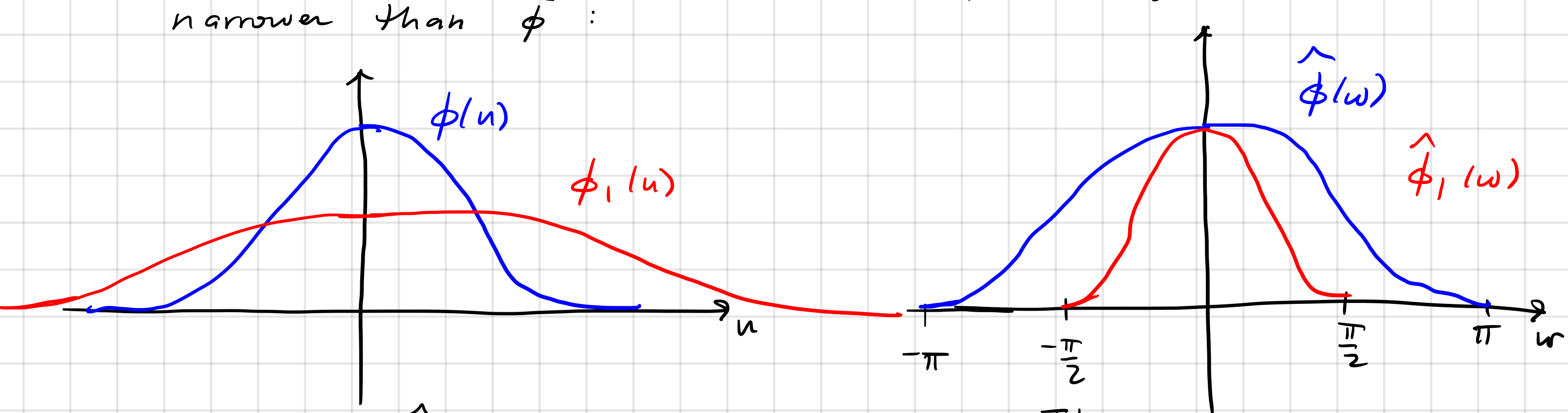



Notice the low pass filter is quite small, and is essentially a $3 \times 3$ filter (as in the VGG network). Because it is a low pass filter, it replaces every pixel in the image with a weighted average of the pixels in a $3\times 3$ neighborhood around the central pixel. The resulting image is similar to the original image, but is a slightly smoothed / blurred version. It retains most of the frequency content of the original image, but loses some of the high frequencies. If one looks carefully at the two images, one can see some of the very fine detail is lost.

We can dilate $\phi$ to enlarge it, which will allow us to smooth the signal more drastically:

$$\phi_J(u) = 2^{-J}\phi(2^{-J}u) \Rightarrow \widehat{\phi}_J(\omega) = \widehat{\phi}(2^J\omega)$$

For $J > 0$, $\phi_J$ will be wider than $\phi$ but $\widehat{\phi}_J$ will be narrower than $\widehat{\phi}$:

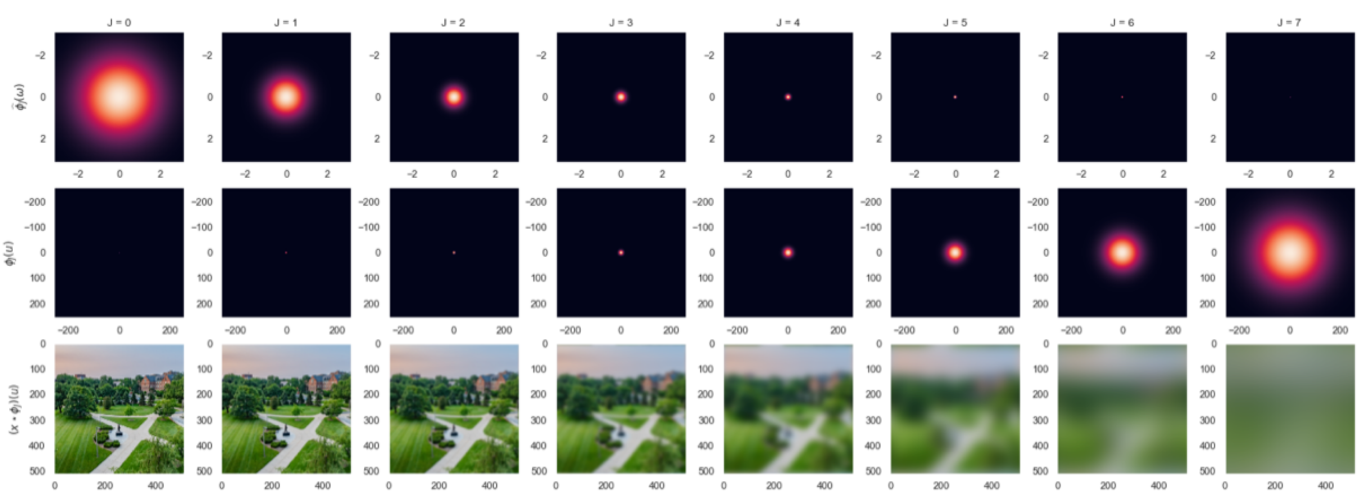



Notice: $\widehat{\phi}_J(\omega) = 0$ for all $|\omega| > \pi/2^J$

When we filter $x$ with $\phi_J$, that is we compute $x * \phi_J$, we blur $x$ even more. Indeed, since

$$\widehat{(x * \phi_J)}(\omega) = \widehat{x}(\omega)\,\widehat{\phi}_J(\omega)$$

we see that $x * \phi_J$ only retains the frequencies of $x$ contained in $|\omega| \leq \pi/2^J$. Here is the filtering of the same image as before for different dyadic scales $2^J$:



The top row is the Fourier transform of $\phi_J$, $\hat{\phi}_J(\omega)$. The middle row is $\phi_J(u)$. The bottom row is $(x * \phi_J)(u)$. The scales range over $0 \le J \le 7$. The low pass function here is a Gaussian,

$$\phi(u) = \frac{1}{2\pi\sigma^2} e^{-|u|^2/2\sigma^2} \Rightarrow \hat{\phi}(\omega) = e^{-\sigma^2|\omega|^2/2}$$

We choose $\sigma = 3/4$. Notice as the scale increases, $\phi_J$ becomes larger and $\hat{\phi}_J(\omega)$ becomes smaller. We average in larger and larger neighborhoods, which progressively blurs the image more and more. From a frequency perspective, we retain fewer and fewer frequencies in the original image $x$. Visually, the increased blur makes it harder to distinguish translations and small deformations of the image. The following theorem quantifies this for translations:

**Theorem (Mallat 2012):** There is a constant $C > 0$, depending on $\phi$, such that for all $t \in \mathbb{R}$ and $x \in L^2(\mathbb{R})$:

$$\| x * \phi_J - x_t * \phi_J \|_2 \le C \cdot 2^{-J} \cdot |t| \cdot \|x\|_2$$

This theorem shows the representation $\Phi(x) = x * \phi_J$ is translation invariant up to the scale $2^J$. But how does this relate to neural networks? To understand this, we will need to appeal to results from sampling theory:

**Theorem (Shannon-Nyquist):** Suppose $\hat{x}(\omega) = 0$ for all $|\omega| > \pi/s$ for some $s > 0$. Then $x$ can be recovered from the downsampled version of $x$ defined by

$$x_d(n) = x(sn), \quad n \in \mathbb{Z}$$

Notice if $s = 1$ then $\hat{x}(\omega) = 0$ for all $|\omega| > \pi$ and we can recover $x : \mathbb{R} \to \mathbb{R}$ from $x_d : \mathbb{Z} \to \mathbb{R}$, $x_d(n) = x(n)$. This is one way to think of a natural image. The underlying scene is $x$ and the image is $x_d$, which has been sampled along "integer" pixels. Since high resolution images are good representations of the scene, we can interpret this as $\hat{x}(\omega) = 0$ for $|\omega| > \pi$ (warning: If you are comparing different cameras, there is some danger in this)

Since we assumed $\hat{\phi} = 0$ for $|\omega| > \pi$, this is why $x * \phi$, depicted earlier, is a good approximation of $x$ since it retains nearly all of $\hat{x}(\omega)$ (only the corners are lost). On the other hand, this is intuitively clear since $\phi$ averaged over a $3\times3$ window.

Notice that for $\phi_J(u) = 2^{-J}\phi(2^{-J}u) \Rightarrow \widehat{\phi_J}(\omega) = \hat{\phi}(2^J\omega)$ we have $\widehat{\phi_J}(\omega) = 0$ for all $|\omega| > \pi/2^J$. Since $\widehat{(x * \phi_J)}(\omega) = \hat{x}(\omega)\widehat{\phi_J}(\omega)$ this means that $\widehat{(x * \phi_J)}(\omega) = 0$ for all $|\omega| > \pi/2^J$. Therefore we can represent $x * \phi_J$ via:

$$(x * \phi_J)_d(n) = (x * \phi_J)(2^J n)$$

Thus we downsample $x * \phi_J$ by a factor $2^J$. This is not quite like CNNs which usually pool in factors of 2. Also, $\phi$ is small, but $\phi_J$ is larger by a factor $2^J$. So there are some differences, at least it would appear so. In fact things are not so different. Indeed the following implements $x * \phi_J$:

$$x \longrightarrow \underbrace{x * \phi_1 \downarrow_2} \longrightarrow (x * \phi_1 \downarrow_2) * \phi_1 \downarrow_2 \longrightarrow \cdots \quad J \text{ times}$$

convolve $x$ with $\phi_1$ (reminder $\hat{\phi}_1(\omega) = 0$ for all $|\omega| > \pi/2$)
and downsample by a factor of 2
Note: $\phi_1$ is essentially $7 \times 7$

Therefore we can implement the translation invariant operator $x * \phi_J$ by composing convolution with $\phi_1$ and downsampling by a factor of 2, $J$ times. This is a simple type of CNN with same single filter at each layer and no nonlinearities.

Okay, so we see that $\Phi(x) = x * \phi_J$ is a translation invariant representation of $x$ and can be viewed as simple CNN. On the other hand, we know

$$\widehat{(x * \phi_J)}(\omega) = \hat{x}(\omega)\hat{\phi}_J(\omega) \neq 0 \text{ only for } |\omega| \leq \pi/2^J$$

So we have lost a lot of $\hat{x}$ and thus $x$ (indeed recall the pictures of $x * \phi_J$ which were very blurry).

To recover the lost information we turn to something called a wavelet transform. A wavelet $\psi: \mathbb{R} \to \mathbb{R}$ or $\psi: \mathbb{R} \to \mathbb{C}$ is a localized, oscillating waveform with zero average. The last property means

$$\hat{\psi}(0) = \int_{\mathbb{R}} \psi(u)\,du = 0$$

Thus, unlike the low pass filter $\phi_J$ for which $\sup_\omega |\hat{\phi}_J(\omega)| = \hat{\phi}_J(0)$, the wavelet $\psi$ has its frequency support concentrated around a frequency (or frequencies) away from zero. Like the low pass filter $\phi$, we can dilate $\psi$:

$$\psi_j(u) = 2^{-j}\psi(2^{-j}u) \Rightarrow \widehat{\psi_j}(\omega) = \hat{\psi}(2^j\omega)$$

A wavelet transform computes:

$$J \geq 1, \qquad W_J x = \{ x * \phi_J(u),\ x * \psi_j(u) \ :\ u \in \mathbb{R},\ 1 \leq j \leq J \}$$

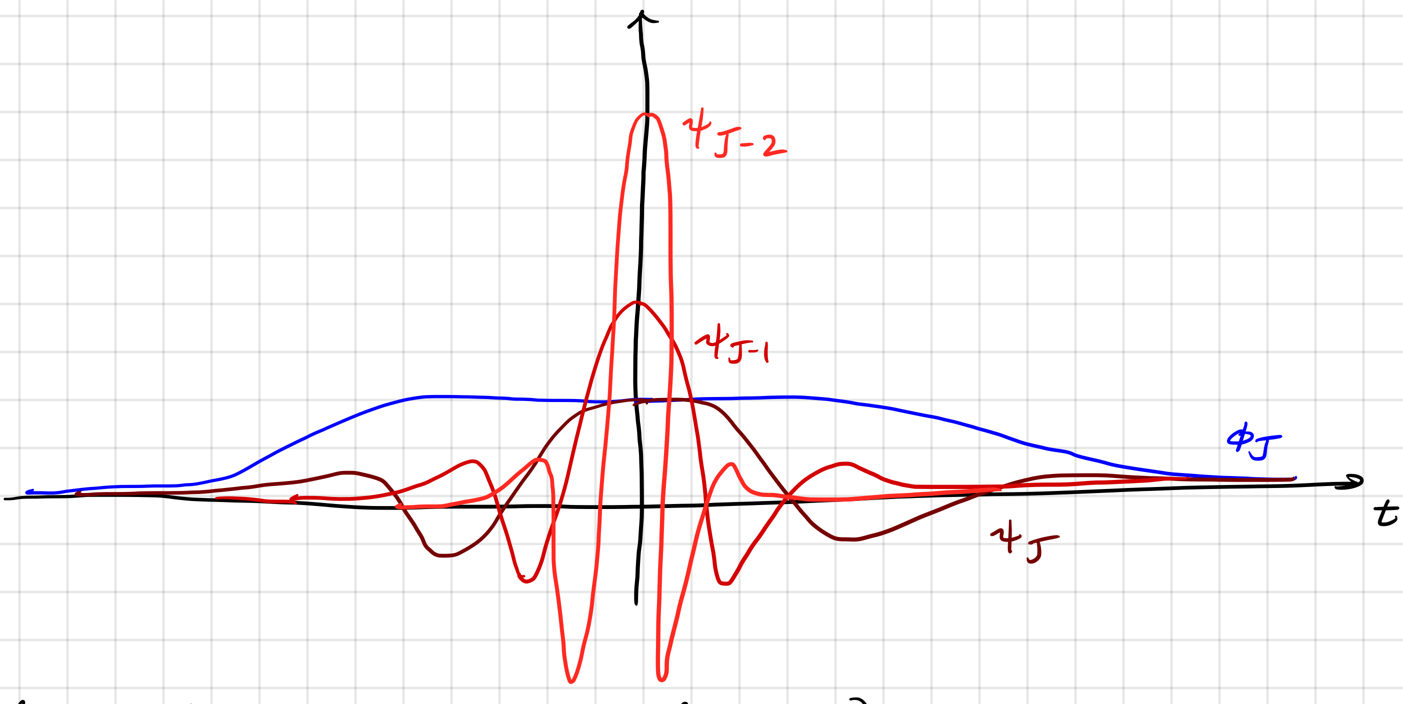
In other words, in addition to averaging over $x$ with $x * \phi_J$, we filter $x$ with $J$ smaller wavelets that recover the details in $x$ lost by $x * \phi_J$. In terms of frequencies, $x * \phi_J$ keeps the low frequencies of $x$ (hence $\phi_J$ is a low pass filter) while $\{x * \psi_j\}_{1 \le j \le J}$ keeps the high frequencies of $x$ (hence the $\psi_j$ filters are called high pass filters).

Suppose, as we observed for natural images, that $\hat{x}(\omega) = 0 \ \forall \ |\omega| > \pi$. If

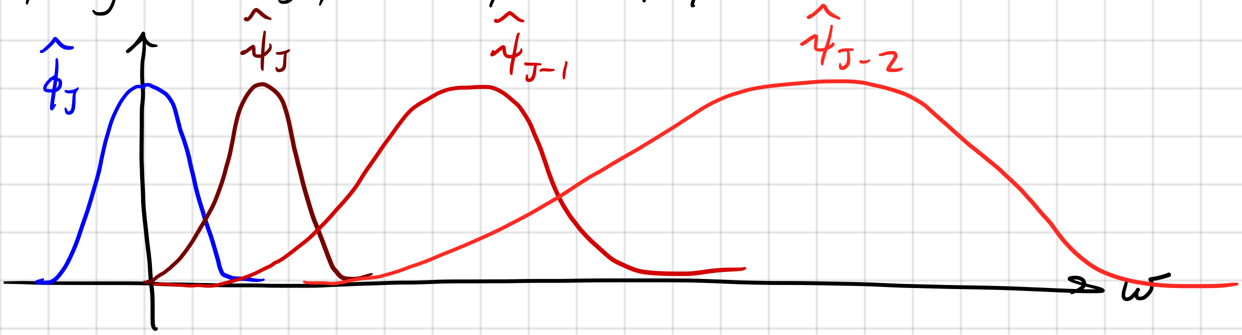
$$\underbrace{0 < A \le |\hat{\phi}_J(\omega)|^2 + \sum_{j=1}^{J} |\hat{\psi}_j(\omega)|^2 \le B < +\infty}_{\text{This means all the frequencies are covered by our low pass } \phi_J \text{ and wavelets } \{\psi_j\}_{1 \le j \le J}} \quad \text{for all } \omega \in [-\pi, \pi]$$

then $W_J x = \{x * \phi_J, x * \psi_j : 1 \le j \le J\}$ is invertible, meaning knowing $W_J x$ is as good as knowing $x$. The proof of this is based on the fact that we stated earlier, which is that knowing $\hat{x}(\omega)$ is as good as knowing $x(u)$.

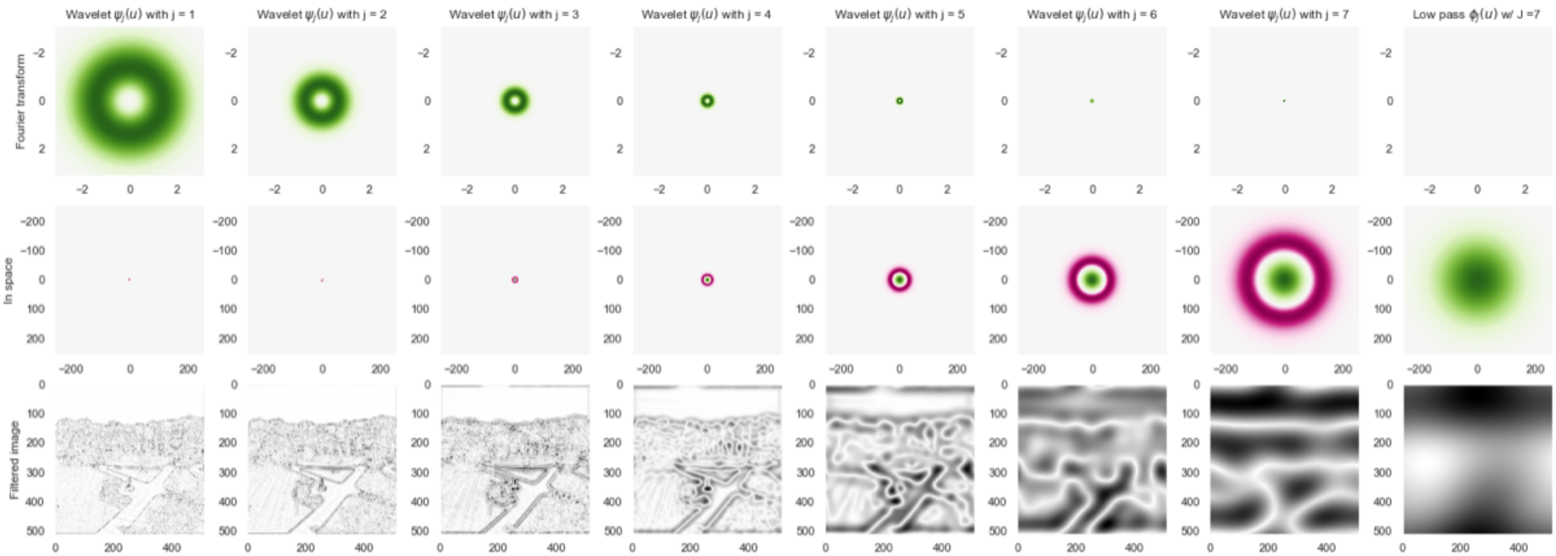
In time/space we have the following plots:




And in frequency (I only plot the positive frequencies):



Here are pictures in 2D on the same image as before:




In the first 2 rows green is positive, pink is negative, white is zero.
In that last row white is zero and black in max positive value.
The first seven columns are wavelets going from small scale in space to large scale in space, so $\psi_j(u)$ (2nd row) and $\hat{\psi}_j(\omega)$ (1st row) for $1 \leq j \leq J = 7$. The last column is the low pass filter $\phi_J(u)$ (2nd row) and $\hat{\phi}_J(\omega)$ (1st row).
We see in frequency the wavelets capture the high frequencies that the low pass misses. These wavelets are localized oscillating waveforms where the oscillations flow radially out of the center.
When computing the filtration $x * \psi_j$ (the 3rd row), the small wavelets act as edge detectors; here we plot:

$$\left[\, |x_r * \psi_j|^2 + |x_g * \psi_j|^2 + |x_b * \psi_j|^2 \right]^{1/2}$$

The larger wavelets capture larger scale information in the image.
In this example, $\phi$ is the same as before and

$$\Delta = \text{Laplacian} \longrightarrow \psi(u) = -(\Delta g)(u), \quad g(u) = \frac{1}{2\pi\alpha^2} e^{-|u|^2/2\alpha^2}, \quad \alpha = \frac{1}{2}$$

$$\Rightarrow \hat{\psi}(\omega) = |\omega|^2 \hat{g}(\omega), \quad \hat{g}(\omega) = e^{-\alpha^2|\omega|^2/2}$$

Like $\phi_J$, we can also implement $x * \psi_j$ with a simple CNN:

$$\underbrace{x \mapsto x * \phi_1 \downarrow_2 \mapsto (x * \phi_1 \downarrow_2) * \phi_1 \downarrow_2 \mapsto \cdots \mapsto (((\underbrace{x * \phi_1 \downarrow_2) * \phi_1 \downarrow_2) * \cdots * \phi_1 \downarrow_2}_{j-1 \text{ times}}) * \psi_1}_{x * \psi_j}$$

where $\phi_1$ and $\psi_1$ are very small filters.