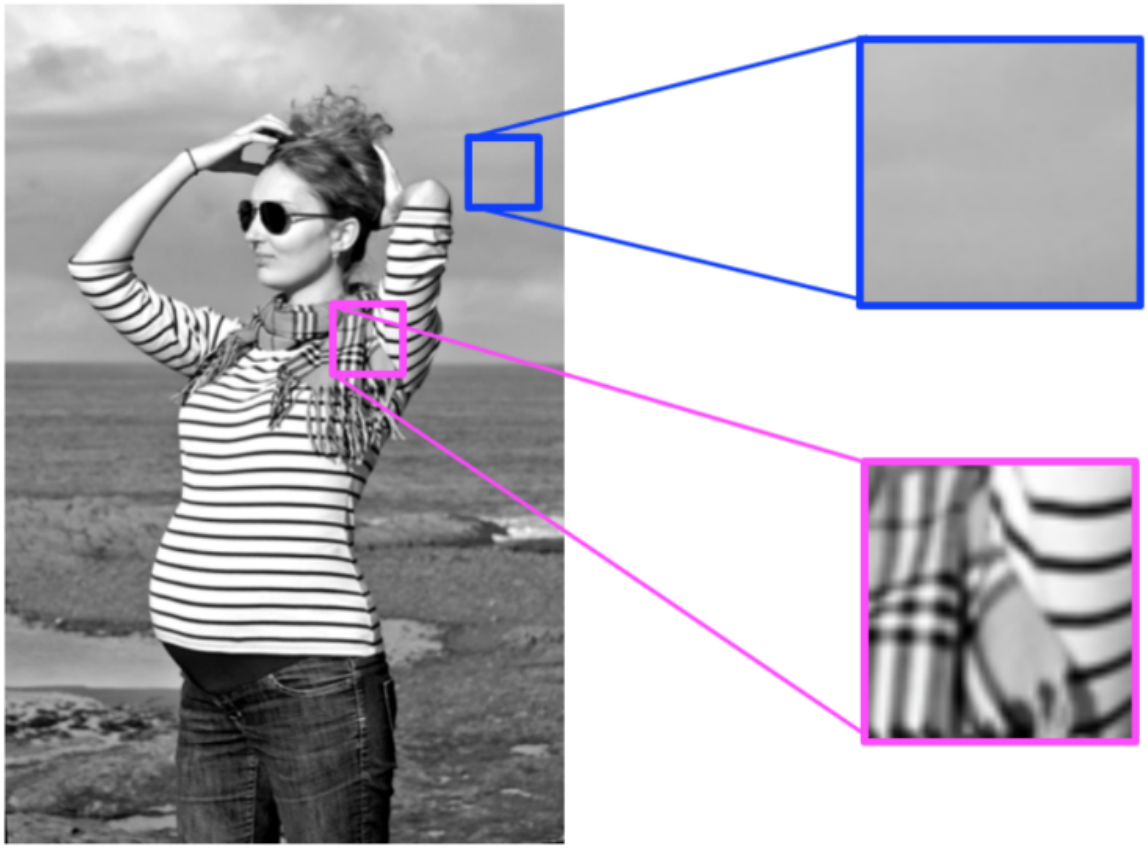
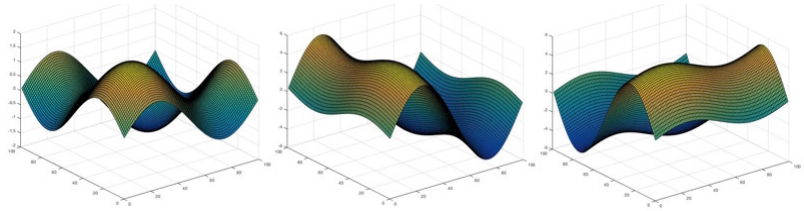
FREQUENCY IN IMAGES

- Similarly, frequency in images is a **rate of change**. But, what does it means for an image to change? Well, images change in space, and a high frequency image is one where the intensity changes a lot. And the level of brightness changes quickly from one pixel to the next
- A low frequency image may be one that is relatively uniform in brightness or changes very slowly. This is easiest to see in an example.
-

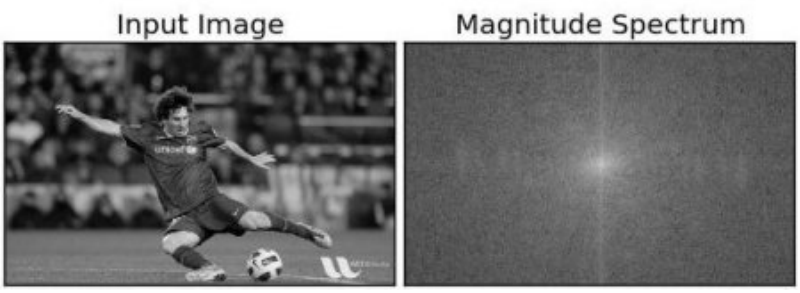


- Most images have both high-frequency and low-frequency components. In the image above, on the scarf and striped shirt, we have a high-frequency image pattern; this part changes very rapidly from one brightness to another
- Higher up in this same image, we see parts of the sky and background that change very gradually, which is considered a smooth, low-frequency pattern.
- **High-frequency components also correspond to the edges of objects in images**, which can help us classify those objects.
- FOURIER TRANSFORM https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_transforms/py_fourier_transform/py_fourier_transform.html
    - The Fourier Transform (FT) is an important image processing tool which is used to decompose an image into its frequency components.

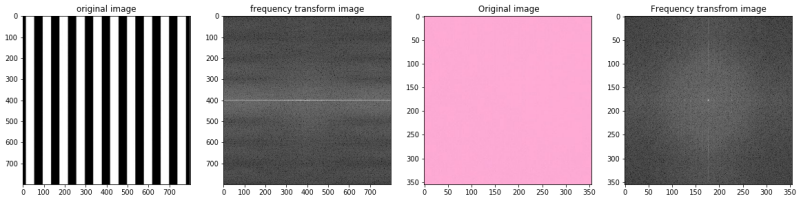
- Fourier Transform is used to analyze the frequency characteristics of various filters. For images, **2D Discrete Fourier Transform (DFT)** is used to find the frequency domain. A fast algorithm called **Fast Fourier Transform (FFT)**
- if the amplitude varies so fast in short time, you can say it is a high frequency signal.
-



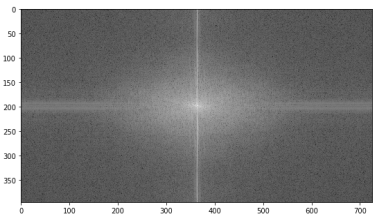
- Fourier Transform in Numpy
  - https://plus.maths.org/content/fourier-transforms-images
  - First we will see how to find Fourier Transform using Numpy. Numpy has an FFT package to do this. **np.fft.fft2()** provides us the frequency transform which will be a complex array. Its first argument is the input image, which is grayscale
  - second argument is optional which decides the size of output array. If it is greater than size of input image, input image is padded with zeros before calculation of FFT. If it is less than input image, input image will be cropped. If no arguments passed, Output array size will be same as input.
-



-



- Low frequencies are at the center of the frequency transform image.
- The transform images for these example show that the solid image has most low-frequency components (as seen by the center bright spot).
- The stripes transform image contains low-frequencies for the areas of white and black color and high frequencies for the edges in between those colors. The stripes transform image also tells us that there is one dominating direction for these frequencies; vertical stripes are represented by a horizontal line passing through the center of the frequency transform image.
-

- Notice that this image has components of all frequencies. You can see a bright spot in the center of the transform image, which tells us that a large portion of the image is low-frequency; this makes sense since the body of the birds and background are solid colors. The transform image also tells us that there are **two** dominating directions for these frequencies; vertical edges (from the edges of birds) are represented by a horizontal line passing through the center of the frequency transform image, and horizontal edges (from the branch and tops of the birds' heads) are represented by a vertical line passing through the center.
-
- Always remember the vertical is represented by Horizontal line and Horizontal is represented by vertical line.