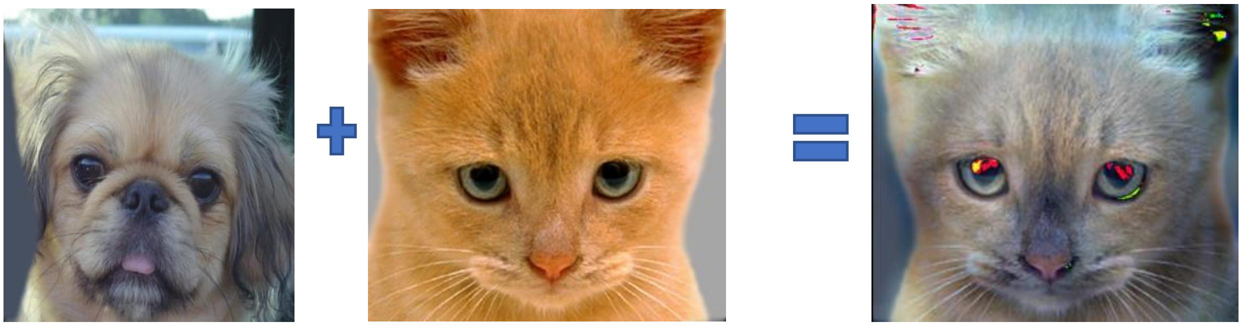
# HYBRID IMAGE

Hybrid image is a digital illusion image that can be developed using two look alike pictures by blurring one image blur (Low Pass) and sharpening other (High Pass). To Develop Hybrid Image, we need to blur images and in computer vision we can blur image efficiently using Gaussian Image Filter or Box Filter.

Example of Hybrid Image:



When viewing this hybrid picture from a distance, the viewer can see the dog because it is low pass filtered, and when seeing it from a close distance, the user can see the cat because it is high pass filtered.

In this assignment I have developed a gaussian filter to blur the images and applied it to develop a Hybrid Image.

## How to Create Hybrid Image:

1)Apply Gaussian Filter on both images

Image1 -> Image1+Filter = Filtered_Image_1

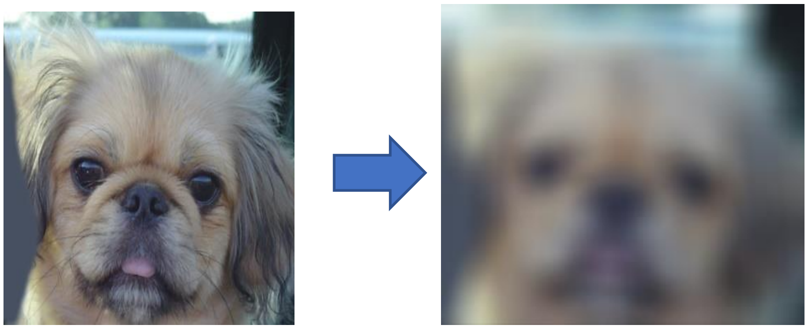
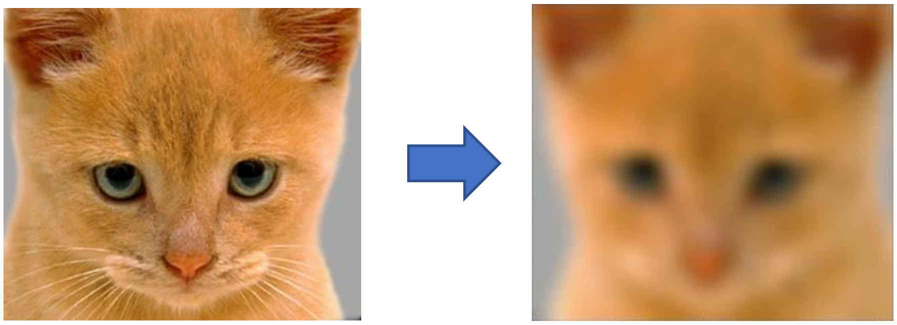


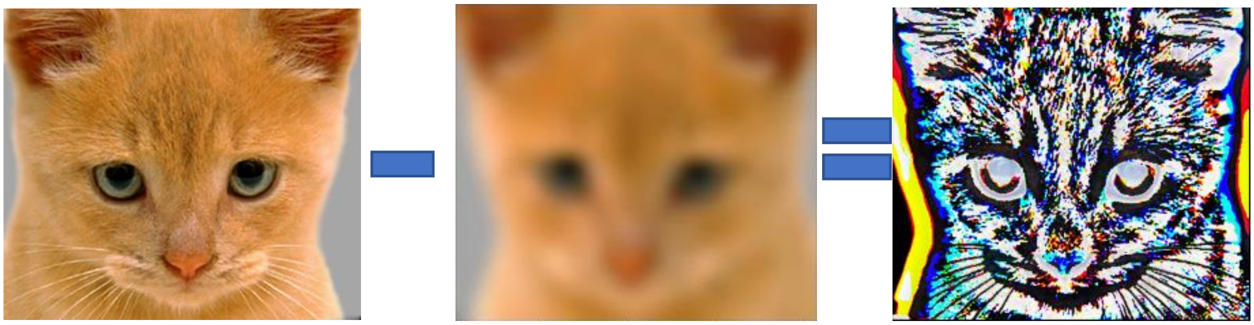
Image2-> Image2+Filter = Filtered_Image_2



2)Filtered Image we got are Low passed, to make Hybrid Image we need to have one image high pass and low pass image.

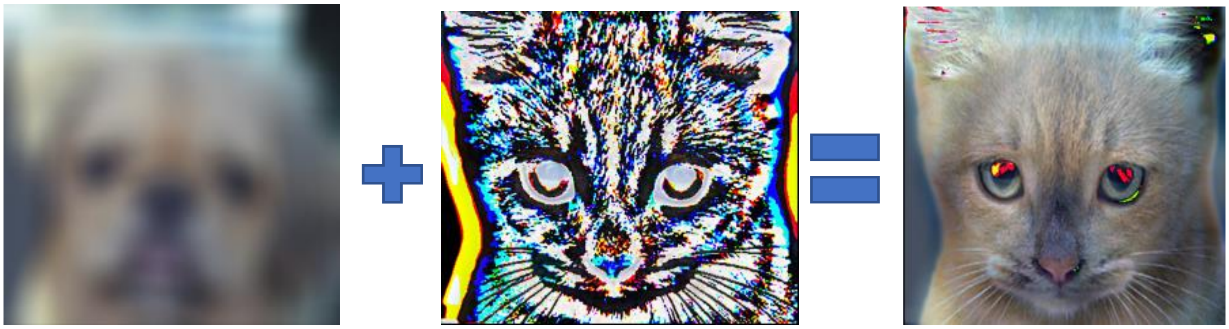
Low_pass_Image1= Filtered_Image_1

High_pass_Image2 = Image2 – Filtered_Image_2



Original – lowpass = Highpass

3) Hybrid Image= Lowpass_Imag1 +Highpass_Imag2



## How to Create Filter?

1) Use OpenCV Python Package to Make Gaussian Filter:

   OpenCV is a Python open source library with various built-in methods that make Image Processing jobs simple.You can simply apply Gaussian filter using cv2 function “cv2.getGaussianKernel(ksize , sigma)” where ksize mean kernel and sigma is simple derivation .

2) Create Own Filter Using Maths Formula:

   Formula to make Gaussian Filter is:

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

We can separate above Formula Into this:

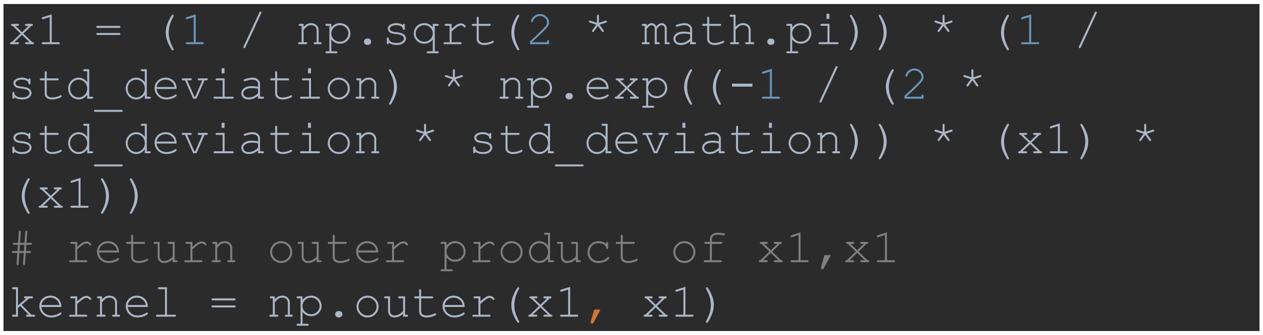
$$\begin{aligned} G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= \left( \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \right) \left( \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{y^2}{2\sigma^2}} \right) \end{aligned}$$

The 2D Gaussian can be expressed as the product of two functions, one a function of *x* and the other a function of *y*

In this case, the two functions are the (identical) 1D Gaussian

We Can Perform separability using Outer Product of Row Metrix and Column Practice.

In My Code I have performed this filter as below:



```
x1 = (1 / np.sqrt(2 * math.pi)) * (1 /
std_deviation) * np.exp((-1 / (2 *
std_deviation * std_deviation)) * (x1) *
(x1))
# return outer product of x1,x1
kernel = np.outer(x1, x1)
```

## How to Apply A Filter Into Image?

Filtering an Image Is simple, it is just matrix multiplication of two matrix image and filter.  We Can Define it using below formula:

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Here, h[m,n] is an Filtered Image(Output) Metrix point at m and n dimension.

g[k,l] is an Filter Metrix point at k and l dimension

f[m+k,n+l] is an Image Metrix point at m+k ,n+l dimension.

I have implemented this in my code:

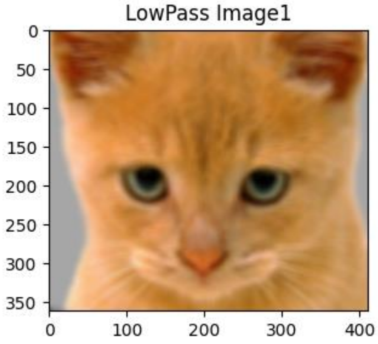
```
for c in range(image.shape[2]):
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            filtered_image[i, j, c] = np.sum(
                np.multiply(padded_image[i:i +
filter_size_w, j:j + filter_size_h, c], filter))
```

Here First Loop is For the RGB image channel, So we have implemented filter three times for each R,G,B Channel.

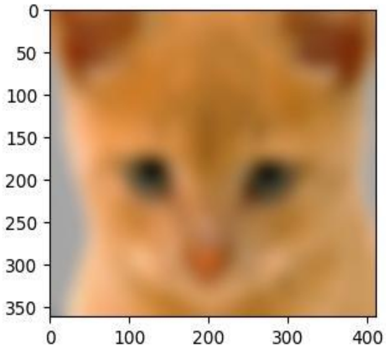
# Results

Low Pass Filter with Different cut off Frequency:

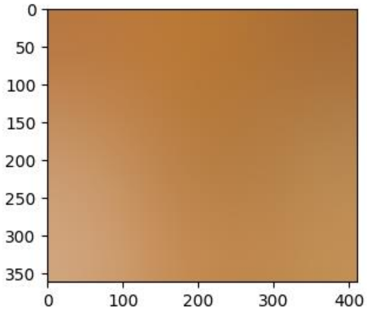
1) Cut -Off Frequency =2



2) Cut-Off Frequency=10



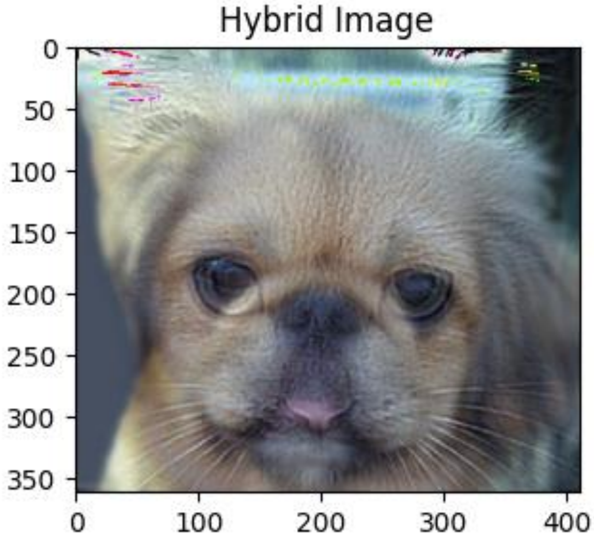
3) Cut-Off Frequency =100



So, when we Increase Cut off Frequency Filter intensity will be increase make image more blur.

**Hybrid Image Results:**

1) Cut-Off Frequency =2



2)Cut-Off Frequency =10