

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ,  
ĐHQGHN

NGÀNH KHOA HỌC MÁY TÍNH

MÔN HỌC: XỬ LÝ ẢNH

---

## Bài tập số 2

---

*Học viên:*

Mai Ngọc Duy –  
22028255

# 1 Giới thiệu

Đây là báo cáo về bài tập thứ 2 của môn Xử Lý Ảnh, lớp INT3403E 20.

## 1.1 Image filtering

### 1.1.1 Hàm padding img, mean filter, median filter

```
def padding_img(img, filter_size=3):
    pad_width = filter_size // 2
    padded_img = cv2.copyMakeBorder(img, pad_width, pad_width,
                                     pad_width, pad_width,
                                     cv2.BORDER_REPLICATE)

    return padded_img

def mean_filter(img, filter_size=3):
    padded_img = padding_img(img, filter_size)
    kernel = (np.ones((filter_size, filter_size),
                      dtype=np.float32) / (filter_size ** 2))
    smoothed_img = cv2.filter2D(padded_img, -1, kernel)
    pad_width = filter_size // 2
    smoothed_img = smoothed_img[pad_width:-pad_width,
                                pad_width:-pad_width]
    return smoothed_img

def median_filter(img, filter_size=3):
    padded_img = padding_img(img, filter_size)
    smoothed_img = np.zeros_like(img)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            smoothed_img[i, j] = np.median(padded_img[i:i
                                                         + filter_size,
                                                         j:j + filter_size])

    return smoothed_img
```

### 1.1.2 Hàm Peak-Signal-to-Noise Ratio (PSNR)

```
def psnr(gt_img, smooth_img):  
    mse = np.mean((gt_img - smooth_img) ** 2)  
    if mse == 0:  
        return float('inf')  
    max_pixel = 255.0  
    psnr_score = 20 * math.log10(max_pixel  
                                  /math.sqrt(mse))  
    return psnr_score
```

### 1.1.3 Nên lựa chọn Median hay Mean filter?

Dựa vào kết quả điểm số PSNR khi cho cả 2 filter xử lý cùng một ảnh bị noise thì em có kết luận như sau:

1. Median filter hiệu quả trong việc loại bỏ hình ảnh bị noise, đồng thời bảo toàn các chi tiết và cạnh của hình ảnh tốt hơn.
2. Còn mean filter lại hiệu quả trong việc loại bỏ nhiễu gaussian nhưng lại khiến ảnh giảm chi tiết và bị mờ một số cạnh.

```
PSNR score of mean filter: 31.637451534808143  
PSNR score of median filter: 37.11957830085524
```

Figure 1: Kết quả điểm số PSNR khi cho median và mean filter xử lý ảnh bị noise

## 1.2 Fourier Transform

### 1.2.1 Frequency Removal Procedure

. Dưới đây là hàm filter frequency theo cách làm của em:

```
def filter_frequency(orig_img, mask):  
  
    f_img = np.fft.fft2(orig_img)
```

```

f_img = np.fft.fftshift(f_img)

f_img = f_img * mask

f_img = np.fft.ifftshift(f_img)

img = np.fft.ifft2(f_img)

f_img = np.fft.ifftshift(f_img)

img = np.abs(img)

f_img = np.abs(f_img)

return f_img, img

```

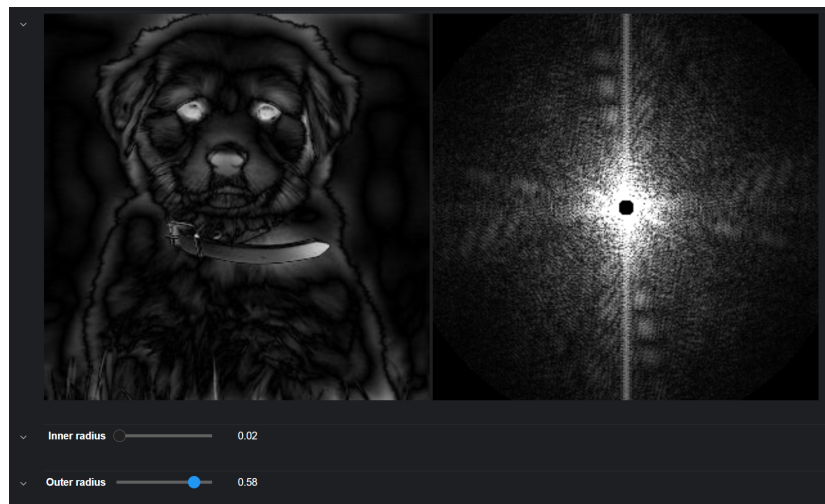


Figure 2: Đây là kết quả của hàm .

### 1.2.2 Hybrid Image

. Hàm để phục vụ việc ảnh tạo ra ảnh hybrid như sau:

```

def create_hybrid_img(img1, img2, r):

```

```

f_img1 = np.fft.fft2(img1)
f_img2 = np.fft.fft2(img2)

f_img1 = np.fft.fftshift(f_img1)
f_img2 = np.fft.fftshift(f_img2)

H, W = img1.shape
x = np.arange(W)
y = np.arange(H)
xv, yv = np.meshgrid(x, y)
xv = xv - W / 2
yv = yv - H / 2
r2 = r * r
mask = (xv**2 + yv**2) <= r2

f_img = mask * f_img1 + (1 - mask) * f_img2

f_img = np.fft.ifftshift(f_img)

img = np.fft.ifft2(f_img)

img = np.real(img)

return img

```

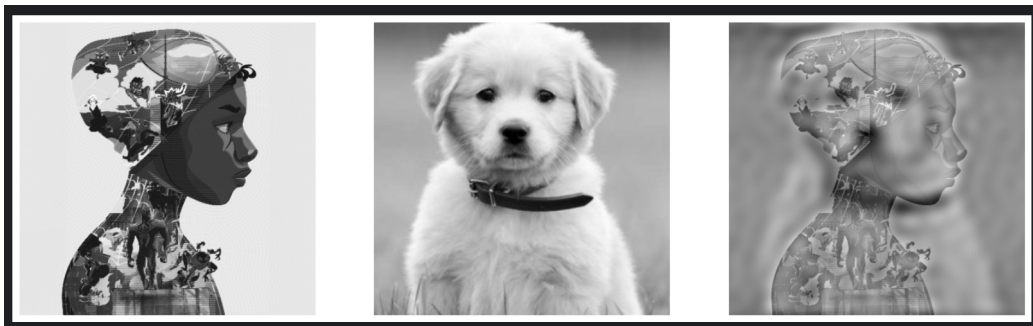


Figure 3: Kết quả của hàm tạo ảnh hybrid.