

Task 4 Unsharp Masking /High-boost Filtering

In [1]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
import urllib
```

In [2]:

```
def show_image(image, title):
    plt.figure(figsize=(10, 10))
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.title(title), plt.xticks([]), plt.yticks([])
    plt.show()
```

In [3]:

```
def plot_images(images, titles, rows, cols , info = -1):
    fig = plt.figure(figsize=(20, 10))
    for i in range(rows * cols):
        ax = fig.add_subplot(rows, cols, i + 1)
        if info != -1:
            ax.text(0.5, 0.5, info[i], fontsize=10, color='black', ha='left', va='top')
        ax.imshow(cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB))
        ax.set_title(titles[i])
        ax.axis('off')
    plt.show()
```

In [4]:

```
req = urllib.request.urlopen('https://github.com/Athimeta/FRA321/blob/main/IMGpro/img/Pokemon.png')
arr = np.asarray(bytearray(req.read()), dtype=np.uint8)
img = cv2.imdecode(arr, -1)
```

Task Start

In [5]:

```
kernel_size = 15
```

Blur the image

Gaussian Blur

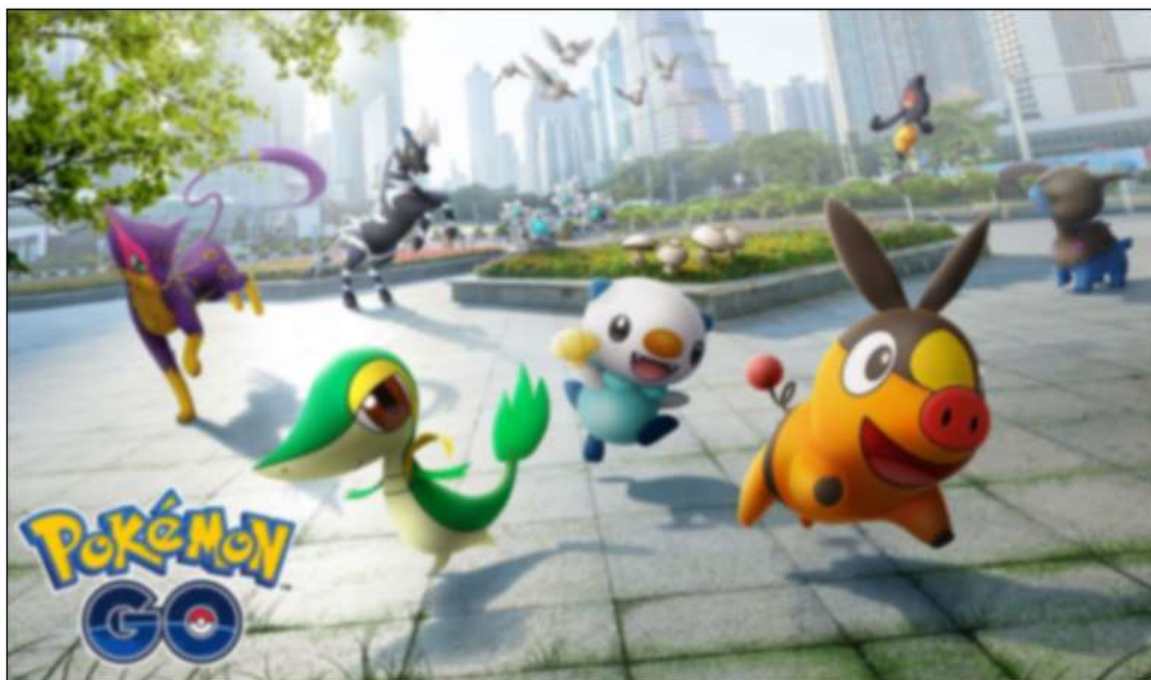
In [6]:

```
img_guassian = cv2.GaussianBlur(src=img, ksize=(kernel_size, kernel_size), sigmaX=0.0)
```

In [7]:

```
show_image(img_guassian, 'Guassian Blur')
```

Guassian Blur



Averaging Filter

In [8]:

```
img_Averaging= cv2.blur(src=img, ksize=(kernel_size, kernel_size))
```

In [9]:

```
show_image(img_Averaging, 'Averaging Blur')
```

Averaging Blur



Subtract the blurred image from the original image

สร้างภาพ Blur ด้วย Gaussian Blur และ Averaging Filter เพื่อทดสอบการทำ Unsharp Masking ด้วย filter ที่แตกต่างกัน

In [10]:

```
mask_guassian = cv2.addWeighted(src1=img,alpha=1,src2=img_guassian,beta=-1,gamma=0)
```

In [11]:

```
show_image(mask_guassian, 'Guassian Mask')
```

Guassian Mask



In [12]:

```
mask_Averaging = cv2.addWeighted(src1=img,alpha=1,src2=img_Averaging,beta=-1,gamma=0)
```

In [13]:

```
show_image(mask_Averaging, 'Averaging Mask')
```

Averaging Mask



โดยจะสามารถเลือกใช้ Gaussian Blur หรือ Averaging Filter ก็ได้แต่ขอบของรูปภาพที่ได้จาก Averaging Filter จะมีความคมชัดมากกว่า Gaussian Blur

Add the mask back to the original image

โดยจะสามารถแสดงได้จากสมการ

$$g(x,y) = f(x,y) + k(f(x,y) - f'(x,y))$$

$$g(x,y) = (f(x,y) + kf(x,y)) - kf'(x,y)$$

หรือก็คือ

$$g(x,y) = \text{img} + k(\text{img}) - k(\text{img_blur})$$

$$g(x,y) = (1+k)(\text{img}) + (-k)(\text{img_blur})$$

Unsharp Masking

In [14]:

```
k = 1.0
```

In [15]:

```
img_sharpened_guassian = cv2.addWeighted(src1=img,alpha=1.0 + k ,src2=img_guassian,beta=-k,
```

In [16]:

```
img_sharpened_Averaging = cv2.addWeighted(src1=img,alpha=1.0 + k,src2=img_Averaging,beta=-k
```

High-boost Filtering

โดยจะกำหนด k ที่ใช้ในการคูณกับ mask ให้เป็น 5.0 เพื่อให้ได้ผลลัพธ์ที่มีความแตกต่างกับภาพต้นฉบับมากขึ้น

In [17]:

```
kh = 5.0
```

In [18]:

```
img_highboost_guassian = cv2.addWeighted(src1=img,alpha=1.0 + kh ,src2=img_guassian,beta=-k
```

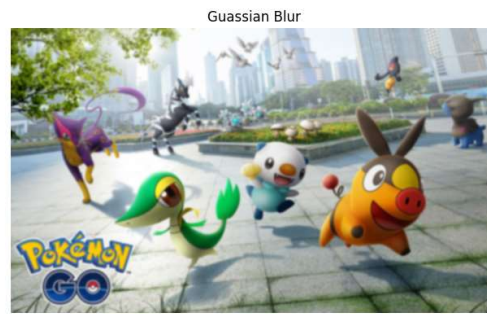
In [19]:

```
img_highboost_Averaging = cv2.addWeighted(src1=img,alpha=1.0 + kh,src2=img_Averaging,beta=-k
```

Display the result

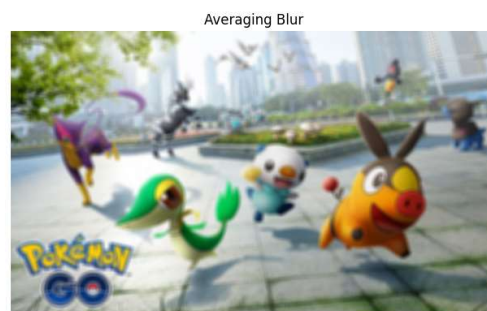
In [20]:

```
plot_images([img, img_guassian,img_sharpened_guassian,img_highboost_guassian], ['Original',
```



In [21]:

```
plot_images([img, img_Averaging,img_sharpened_Averaging,img_highboost_Averaging], ['Original',
```



Conclusion

จากการลองทำ Unsharp Masking และ High-boost Filtering จะเห็นว่าหลังจากการทำ Unsharp Masking จะทำให้ภาพมี

ขอบที่คมชัดมากขึ้นกว่าภาพต้นฉบับและ High-boost Filtering จะทำให้ภาพมีขอบที่ยิ่งคมชัดมากกว่า ซึ่งเป็นไปตามทฤษฎีที่เราได้เรียนมา โดยการใช้ Gaussian Filter หรือ Averaging Filter ในการทำ Blur ให้ผลลัพธ์ที่ค่อนข้างคล้ายกันในการทำ Unsharp Masking แต่ในการทำ High-boost Filtering Averaging Filter จะทำให้ผลลัพธ์ที่ได้มีขอบที่มีความคมชัดมากกว่า Gaussian Filter