**Comment**

實驗結果顯示，**平均濾波（Average Filter）** 相較於 **中值濾波（Median Filter）**，對影像的模糊化效果更為顯著。當 **核大小 (n) 增加時**，濾波範圍擴大，使得影像變得更加模糊。這是因為平均濾波對影像中的所有像素取均值，會平滑掉邊緣與細節資訊，導致整體畫面變得柔和。

相對而言，中值濾波雖然也能減少雜訊，但其機制是選取局部區域內的中值，因此能夠較好地保留影像的邊緣與細節，特別適用於去除Salt-and-Pepper Noise。

此外，在**未銳化遮罩（Unsharp Masking）** 的實驗中，**增大銳化係數 (k)** 會使得細節增強效果更加明顯。這是因為 k 會影響增強邊緣的強度，當 k 值較大時，原圖與模糊影像的差異被放大，從而強化邊緣與紋理細節，使影像看起來更加銳利。

總結來說，**核大小 (n) 控制模糊程度，銳化係數 (k) 影響細節增強強度**，在影像處理中需根據不同應用場景調整這些參數，以達到最佳的影像處理效果。

**Photos**

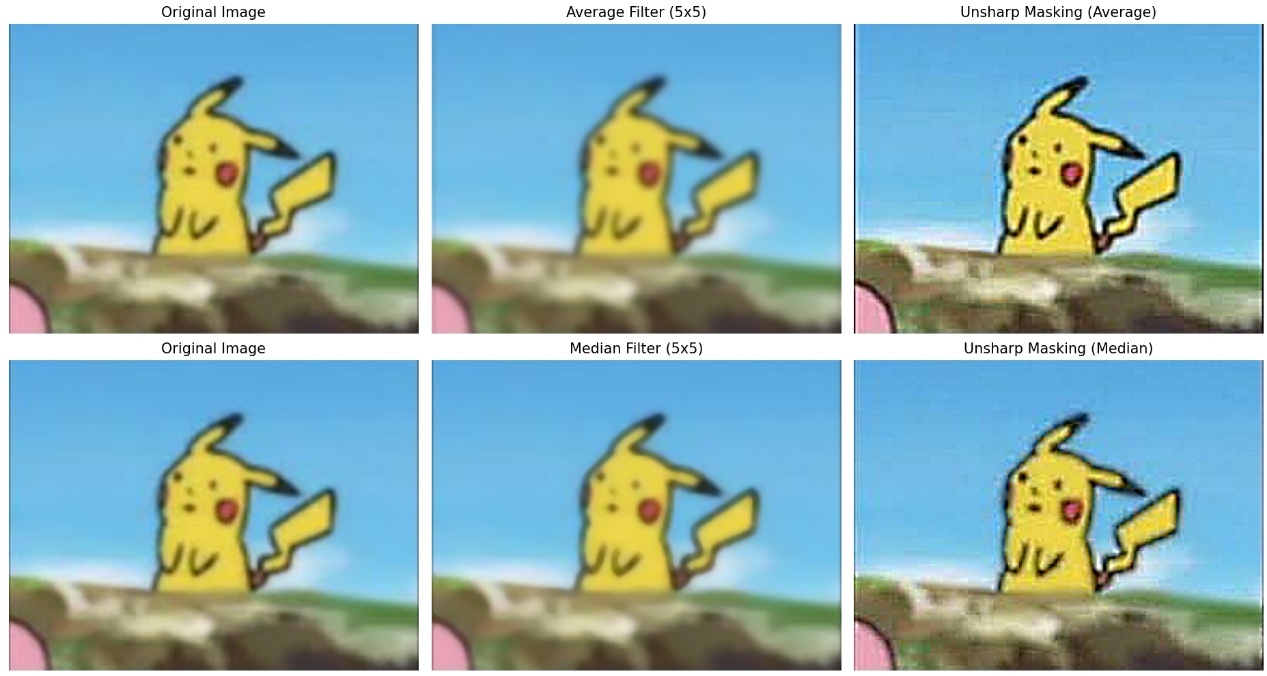
Origin



N = 5, k = 1.5



N = 5, k = 3.0



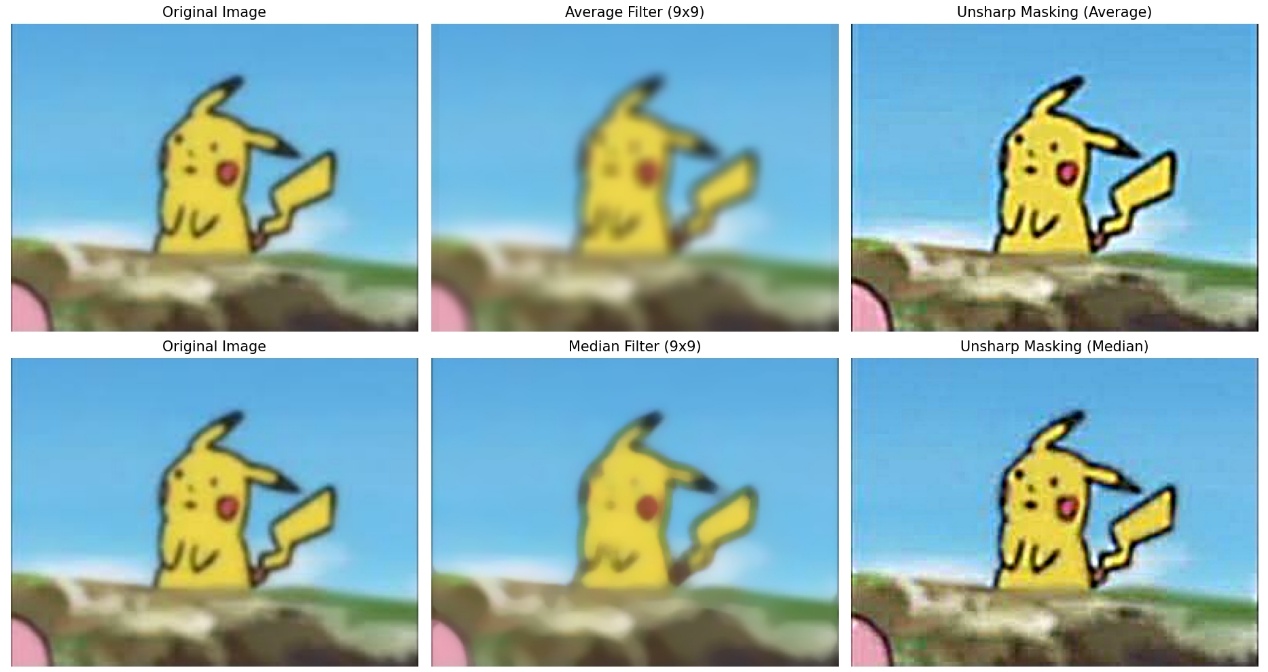
N = 7, k = 1.5



N = 7, k = 3.0



N = 9, k = 1.5



N = 9, k = 3.0



**Program**

import cv2

import numpy as np

import matplotlib.pyplot as plt

image = cv2.imread('images.jpg', cv2.IMREAD\_COLOR)

n = 5

# (a) 應用平均濾波器

average\_filter\_image = cv2.blur(image, (n, n))

# (b) 應用媒體濾波器

median\_filter\_image = cv2.medianBlur(image, n)

image\_int16 = image.astype(np.int16)

average\_filter\_image\_int16 = average\_filter\_image.astype(np.int16)

median\_filter\_image\_int16 = median\_filter\_image.astype(np.int16)

average\_image = cv2.subtract(image\_int16, average\_filter\_image\_int16)

median\_image = cv2.subtract(image\_int16, median\_filter\_image\_int16)

k = 3.0

sharpened\_average\_image = np.clip(image\_int16 + k \* average\_image, 0, 255).astype(np.uint8)

sharpened\_median\_image = np.clip(image\_int16 + k \* median\_image, 0, 255).astype(np.uint8)

plt.figure(figsize=(15, 12))

plt.subplot(2, 3, 1)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

plt.title("Original Image")

plt.axis('off')

plt.subplot(2, 3, 2)

plt.imshow(cv2.cvtColor(average\_filter\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Average Filter ({n}x{n})")

plt.axis('off')

plt.subplot(2, 3, 3)

plt.imshow(cv2.cvtColor(sharpened\_average\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Unsharp Masking (Average)")

plt.axis('off')

plt.subplot(2, 3, 4)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

plt.title("Original Image")

plt.axis('off')

plt.subplot(2, 3, 5)

plt.imshow(cv2.cvtColor(median\_filter\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Median Filter ({n}x{n})")

plt.axis('off')

plt.subplot(2, 3, 6)

plt.imshow(cv2.cvtColor(sharpened\_median\_image, cv2.COLOR\_BGR2RGB))

plt.title("Unsharp Masking (Median)")

plt.axis('off')

plt.tight\_layout()

plt.show()