**Title:** Dilation.

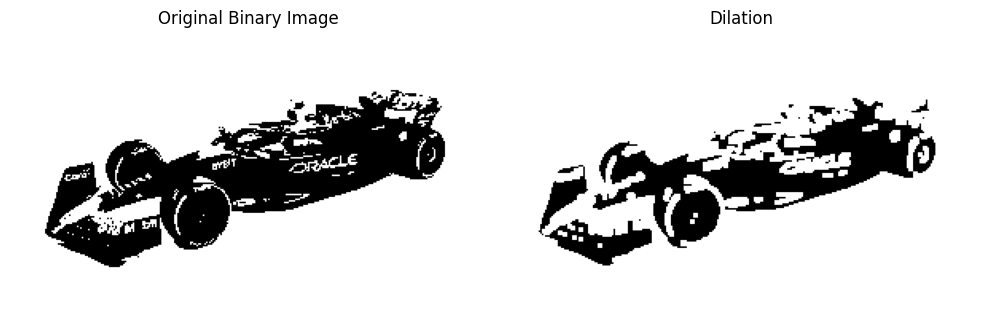
**Objective:** The objective of this code is to perform dilation on a binary image using a 3x3 kernel. Dilation is a morphological operation that expands white regions in the binary image, increasing the size of objects and potentially filling small holes.

**Theory:** Dilation is a morphological operation that works by expanding the boundaries of the foreground (white) pixels in a binary image. For each pixel, the kernel is applied, and if at least one pixel under the kernel is white (255), the central pixel is set to white (255). This operation results in larger white regions and is often used to fill small gaps or connect nearby objects in binary images. In this implementation, the image is padded to handle edge cases, and the kernel is manually applied to each pixel in the image.

Code:

|  |
| --- |
| from PIL import Image  import numpy as np  import matplotlib.pyplot as plt  img = Image.open('formula.jpeg').convert('L')  img\_np = np.array(img)  binary = np.where(img\_np > 127, 255, 0).astype(np.uint8)  kernel = np.ones((3, 3), np.uint8)  p = 1  # *Padding size*  padded = np.pad(binary, ((p, p), (p, p)), mode='constant', constant\_values=0)  # *Padding image*  dilation = np.zeros\_like(binary)  for i in range(p, padded.shape[0] - p):      for j in range(p, padded.shape[1] - p):            region = padded[i - p:i + p + 1, j - p:j + p + 1]              if np.any(region[kernel == 1] == 255):              dilation[i - p, j - p] = 255          else:              dilation[i - p, j - p] = 0  # *Plot both images*  plt.figure(figsize=(10, 4))  plt.subplot(1, 2, 1)  plt.imshow(binary, cmap='gray')  plt.title('Original Binary Image')  plt.axis('off')  plt.subplot(1, 2, 2)  plt.imshow(dilation, cmap='gray')  plt.title('Dilation')  plt.axis('off')  plt.tight\_layout()  plt.show() |

**Output:**



**Conclusion:** This code demonstrates the dilation operation, which enlarges white regions in a binary image by expanding the edges of objects. The operation can be used to fill small holes and connect disjoint objects. This morphological operation is commonly used in tasks such as image enhancement, object detection, and noise reduction.