Visual Information Fidelity (VIF) score between two images: one is a reference image (img\_ref) and the other is a distorted image (img\_dist).

### Key Functions and Steps:

#### 1. integral\_image(x):

* This function computes the integral image (also known as a summed-area table) for an input image x.
* The integral image is a useful preprocessing step for efficient image processing, as it allows for the rapid calculation of sums over rectangular regions in the image.
* Formula: int\_x[i, j] = sum of all pixels (0,0) to (i,j).

#### 2. moments(x, y, k, stride):

* This function computes the first-order (mean), second-order (variance), and covariance moments between the reference image (x) and distorted image (y).
* The moments are calculated using an integral image approach, which allows for the efficient calculation of local statistics in a sliding window.
* It also pads both images to ensure that regions near the edges of the image can still be processed correctly.
* k: Size of the neighborhood window.
* stride: Step size for sliding the window.
* It returns:
  + mu\_x: Mean of x (reference image).
  + mu\_y: Mean of y (distorted image).
  + var\_x: Variance of x.
  + var\_y: Variance of y.
  + cov\_xy: Covariance between x and y.

#### 3. vif\_spatial(img\_ref, img\_dist, k=11, sigma\_nsq=0.1, stride=1, full=False):

* This is the main function where VIF is calculated.
* It takes two images, the reference image (img\_ref) and the distorted image (img\_dist), and calculates the Visual Information Fidelity (VIF) score between them.
* Steps:
  1. Convert images to float32: It converts both the reference and distorted images to 32-bit floating point format (float32) to ensure that calculations can be performed with sufficient precision.
  2. Call moments(): It calculates the moments (mu\_x, mu\_y, var\_x, var\_y, cov\_xy) between the reference and distorted images using a sliding window approach.
  3. Calculate g and sv\_sq:
     + g is the gain factor, calculated as the covariance between the two images divided by the variance of the reference image.
     + sv\_sq is the signal variance in the distorted image, adjusted for the gain factor g.
  4. Zero out small values: If any variance or covariance is close to zero (or negative due to numerical precision issues), the function sets them to zero or adjusts accordingly.
  5. Calculate the VIF score:
     + The VIF score is calculated by summing over the logarithms of the signal-to-noise ratios across all pixels. This is done for each local region of the image.
     + The formula used is based on a ratio of signal power to noise power (sigma\_nsq), incorporating the moments, gain factor g, and signal variance sv\_sq.
  6. Return the VIF score: If full=False, only the final VIF score is returned. If full=True, additional details like intermediate sums are returned.

### Formula Breakdown (VIF Calculation):

The VIF score formula is based on measuring the quality of the image in terms of how much visual information is preserved in the distorted image compared to the reference.

The formula used in the function is:

VIF=∑log(1+σnsq​varx​​)∑log(1+g2⋅sv\_sq+σnsq​varx​​)​

Where:

* g = Gain factor (covariance between the reference and distorted images divided by the variance of the reference image).
* varx​ = Variance of the reference image.
* sv\_sq = Signal variance in the distorted image.
* σnsq​ = Noise variance (a small constant to prevent division by zero).

#### Full vs. Non-Full VIF:

* When full=False, only the final VIF score is returned. This gives a summary of how similar the distorted image is to the reference.
* When full=True, the function also returns intermediate results:
  + The sum of the logarithmic terms for both the numerator and the denominator.
  + The detailed VIF score map showing how the quality varies across the image.

### Example Usage:

1. Read images:
   * The reference image (img\_ref) and distorted image (img\_dist) are loaded using OpenCV's cv2.imread(). These are grayscale images (i.e., single-channel images).
2. Resize images:
   * Both images are resized to ensure they are the same size. This is crucial for VIF, as it compares the pixel values of the reference and distorted images directly.
3. Run the VIF function:
   * vif\_spatial() is called to compute the VIF score for the images with a neighborhood window size (k=11) and noise variance (sigma\_nsq=0.1).
4. Output the result:
   * The VIF score is printed, which tells you how much visual information is preserved in the distorted image relative to the reference.

### Expected Result:

The VIF score quantifies how much of the visual information in the reference image is retained in the distorted image. A higher VIF score indicates that the distorted image is visually similar to the reference, while a lower VIF score indicates more distortion or degradation.

For example, if the images are identical, the VIF score will be close to 1. If the distorted image has significant degradation (e.g., due to compression, noise, or other artifacts), the VIF score will be closer to 0.

Output:

(10x Colon)

|  |  |  |
| --- | --- | --- |
| Input Image | Output Image | VIF score |
| 1 | 2 | 0.012318955942089541 |
| 1 | 3 | 0.012175518521742135 |
| 1 | 4 | 0.012318955942089541 |
| 1 | 5 | 0.01662302110051669 |
| 1 | 6 | 0.016284134867074054 |
| 7 | 8 | 0.01639609932180217 |
| 9 | 10 | 0.016805671935029864 |
| 9 | 11 | 0.017856759503448966 |
| 9 | 12 | 0.015922880619809433 |
| 13 | 14 | 0.0497796749183568 |
| 13 | 15 | 0.9999999998460583 |
| 13 | 16 | 0.04996996481961997 |
| 13 | 17 | 0.054334297278039026 |
| 13 | 18 | 0.054334297278039026 |

|  |  |  |
| --- | --- | --- |
| 19 | 20 | 0.9999999998563928 |
| 19 | 21 | 0.0334483851593107 |
| 19 | 22 | 0.04205262287776285 |
| 19 | 23 | 0.03158162661026976 |
| 19 | 24 | 0.03158162661026976 |
| 25 | 26 | 0.03827071943324314 |
| 25 | 27 | 0.03827071943324314 |
| 25 | 28 | 0.03827071943324314 |
| 25 | 29 | 0.03668437327353689 |
| 25 | 30 | 0.03398924799378815 |
| 25 | 31 | 0.03788515725208807 |
| 32 | 33 | 0.9999999998488189 |
| 32 | 34 | 0.05042186927301609 |
| 35 | 36 | 0.05726499633762726 |