**Image Enhancement**

**Definisi**: Image Enhancement adalah teknik untuk meningkatkan kualitas citra, terutama untuk tujuan aplikasi lanjutan seperti deteksi objek. Teknik ini penting karena seringkali citra mentah memiliki gangguan seperti noise, kurang tajam, terlalu terang/gelap, atau blur akibat pergerakan objek atau kamera.

**Metode perbaikan citra** dikelompokkan menjadi dua berdasarkan ranah operasinya:

* **Ranah Spasial** - Manipulasi langsung pada pixel citra.

1. **Operasi Aras Titik**: Memanipulasi satu pixel tunggal untuk mengubah kecerahan, kontras, atau menghasilkan efek tertentu.
2. **Operasi Aras Lokal**: Menggunakan kelompok pixel bertetangga.
3. **Operasi Aras Global**: Menggunakan seluruh pixel citra.

* **Ranah Frekuensi** - Konversi citra dari ranah spasial ke frekuensi (misalnya, dengan Fourier Transform) lalu manipulasi nilai frekuensi.

**Teknik Pencerahan dan Menegatifkan Citra**

* **Kecerahan Gambar** (Image Brightening): Metode ini mengatur intensitas piksel untuk meningkatkan keseluruhan kecerahan citra. Hal ini berguna dalam memperjelas detail pada citra yang tampak gelap.
* **Citra Negatif** (Image Negatif): Teknik ini membalikkan intensitas piksel, di mana piksel yang terang menjadi gelap dan sebaliknya. Berguna untuk aplikasi medis, seperti memperjelas tulang pada citra X-ray.
* **Peregangan Kontras** (Contrast Stretching): Teknik ini meningkatkan rentang intensitas warna citra sehingga perbedaan antara bagian terang dan gelap menjadi lebih terlihat, membantu memperjelas detail dalam area tertentu.
* **Koreksi Gamma**: Koreksi ini mengatur distribusi intensitas agar lebih sesuai dengan karakteristik perangkat tampilan (monitor, layar ponsel). Koreksi gamma dapat membuat citra lebih jelas di perangkat yang memiliki rentang intensitas berbeda.

**Transformasi Intensitas Grayscale**

* Fungsi Linear - Transformasi langsung tanpa perubahan (identitas) atau invers untuk citra negatif.
* Transformasi Log - Mengompres rentang grayscale; cocok untuk citra dengan rentang dinamis tinggi.
* Transformasi Pangkat - Memodifikasi citra gelap atau terang menggunakan konstanta eksponensial (gamma correction).

**Perbaikan Kontras**

**Contrast Stretching** digunakan untuk meningkatkan rentang grayscale dalam citra kontras rendah. Transformasi ini mengubah pixel di bawah nilai ambang menjadi lebih gelap dan di atas nilai ambang menjadi lebih terang.

**Gray-level dan Bit-plane Slicing**

* Gray-level Slicing - Menonjolkan rentang grayscale tertentu untuk aplikasi khusus, misalnya menyoroti cacat dalam citra medis.
* Bit-plane Slicing - Menguraikan citra menjadi delapan bit untuk analisis detail pada tiap tingkatan bit.

**Pengubahan Histogram Citra:**

**Perataan Histogram**: Mengatur distribusi intensitas agar lebih merata, membuat fitur dalam citra menjadi lebih mudah dilihat dan dianalisis. Cocok untuk citra yang pencahayaannya tidak merata.

**Spesifikasi Histogram**: Mengubah histogram citra agar sesuai dengan histogram tertentu (target). Berguna saat ingin mencocokkan kualitas pencahayaan citra dengan referensi.

**Pelembutan Citra** (Image Smoothing): Smoothing mengurangi noise dalam citra dengan menggunakan filter, seperti filter rata-rata atau Gaussian. Teknik ini menghaluskan detail kecil, tetapi dapat mengaburkan tepi.

**Penajaman Tepi** (Edge Sharpening): Meningkatkan kejelasan tepi objek dalam citra. Penajaman ini dilakukan untuk membuat batas antara objek lebih terlihat jelas, membantu memperjelas bentuk dan kontur dalam citra.

**Pewarnaan Semu** (Pseudocoloring): Memberikan warna pada citra grayscale untuk memperkaya informasi visual. Teknik ini sering digunakan dalam citra medis dan ilmiah untuk membedakan antara bagian yang berbeda dalam gambar.

**Pengubahan Geometrik**: Melakukan perubahan seperti rotasi, skala, atau pemotongan pada citra. Pengubahan ini sering diterapkan untuk menyesuaikan posisi atau orientasi citra sesuai kebutuhan analisis atau tampilan.

1. Import Library

import cv2

import numpy as np

import matplotlib.pyplot as plt

**penjelasan:**

* cv2: Digunakan untuk pengolahan citra, dalam hal ini dari OpenCV.
* numpy: Digunakan untuk perhitungan numerik yang melibatkan array.
* matplotlib.pyplot: Digunakan untuk menampilkan citra hasil pemrosesan.

1. Memuat citra gray scale

image = cv2.imread('/content/drive/MyDrive/Kuliah/CItra Digital/fikri.jpg', cv2.IMREAD\_GRAYSCALE)

**Penjelasan:**

* cv2.imread(): Membaca citra dari direktori yang ditentukan.
* cv2.IMREAD\_GRAYSCALE: Membaca citra dalam format grayscale (hitam putih), sehingga citra hanya memiliki satu kanal warna (kecerahan) dengan nilai antara 0-255.

1. Meningkatkan Kecerahan dengan Penjumlahan

bright\_image\_add = cv2.add(image, 50)

**Penjelasan:**

* cv2.add(): Menambahkan nilai pada setiap piksel dalam citra.
* 50: Konstanta yang ditambahkan pada setiap piksel. Nilai kecerahan setiap piksel akan meningkat sebanyak 50 unit, dengan catatan bahwa nilai maksimal tetap di 255 (overflow otomatis dikendalikan oleh OpenCV). Hasilnya adalah citra yang lebih terang daripada citra asli.

1. Meningkatkan Kecerahan dengan Perkalian

bright\_image\_mult = cv2.multiply(image, 1.2)

Penjelasan:

* cv2.multiply(): Mengalikan setiap nilai piksel dalam citra dengan konstanta yang diberikan.
* 1.2: Faktor yang mengalikan setiap piksel, memperbesar intensitas kecerahan sebesar 20%.

Metode ini juga mencerahkan citra, tetapi hasilnya lebih lembut daripada penambahan karena pengaruh perkalian lebih merata di seluruh rentang piksel.

1. Menampilkan Hasil

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)

plt.title('Original Image')

plt.imshow(image, cmap='gray')

plt.subplot(1, 3, 2)

plt.title('Brightened by Addition')

plt.imshow(bright\_image\_add, cmap='gray')

plt.subplot(1, 3, 3)

plt.title('Brightened by Multiplication')

plt.imshow(bright\_image\_mult, cmap='gray')

plt.show()

**Penjelasan:**

* plt.figure(): Menentukan ukuran tampilan gambar.
* plt.subplot(): Mengatur tata letak gambar menjadi 1 baris dan 3 kolom.
* plt.imshow(): Menampilkan citra dalam format grayscale (cmap='gray').

Hasil ditampilkan dalam tiga gambar berdampingan:

* Gambar asli (grayscale).
* Gambar hasil peningkatan kecerahan dengan metode penjumlahan.
* Gambar hasil peningkatan kecerahan dengan metode perkalian.

**Membuat Citra Negatif**

negative\_image = 255 – image

**Penjelasan:**

* 255 - image: Pada setiap piksel di citra asli, dilakukan operasi pengurangan nilai piksel tersebut dari 255. Hal ini mengubah nilai piksel cerah menjadi gelap, dan sebaliknya.
* Jika piksel pada citra asli bernilai tinggi (dekat 255), misalnya 200, nilai di citra negatif akan menjadi 255 - 200 = 55 (gelap).
* Jika piksel bernilai rendah (dekat 0), misalnya 50, nilai di citra negatif akan menjadi 255 - 50 = 205 (cerah).

Hasilnya adalah citra negatif, di mana area gelap menjadi terang, dan area terang menjadi gelap, mirip dengan film negatif pada fotografi.

**Menampilkan Hasil Citra Asli dan Citra Negatif:**

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title('Original Image')

plt.imshow(image, cmap='gray')

plt.subplot(1, 2, 2)

plt.title('Negative Image')

plt.imshow(negative\_image, cmap='gray')

plt.show()

**Penjelasan:**

* **plt.figure()**: Menentukan ukuran tampilan gambar.
* **plt.subplot()**: Menyusun tampilan menjadi 1 baris dan 2 kolom, dengan citra asli di sebelah kiri dan citra negatif di sebelah kanan.
* **plt.title()**: Memberikan judul pada setiap citra.
* **plt.imshow()**: Menampilkan gambar dalam skala keabuan (cmap='gray').
* **plt.show()**: Menampilkan semua gambar di jendela output.

**Menghitung Nilai Minimum dan Maksimum Piksel:**

rmin, rmax = image.min(), image.max()

**Penjelasan:**

* image.min(): Mencari nilai piksel terkecil pada citra (nilai terendah).
* image.max(): Mencari nilai piksel terbesar pada citra (nilai tertinggi).
* Kedua nilai ini (rmin dan rmax) digunakan untuk menghitung rentang intensitas pada citra asli.

**Melakukan Peregangan Kontras (Contrast Stretching):**

stretched\_image = ((image - rmin) \* (255 / (rmax - rmin))).astype(np.uint8)

**Penjelasan:**

*  **(image - rmin)**: Mengurangi setiap piksel dengan rmin, sehingga nilai terkecil dalam citra menjadi 0.
* **(255 / (rmax - rmin))**: Mengalikan hasil sebelumnya dengan skala yang memetakan rentang nilai citra asli ke rentang nilai penuh (0-255) pada citra 8-bit. Hasil akhir dari penghitungan ini adalah nilai baru yang lebih tersebar pada rentang 0 hingga 255.
* **astype(np.uint8)**: Mengonversi hasil ke tipe uint8, sehingga sesuai untuk citra 8-bit (rentang nilai 0-255).
* **Hasilnya** adalah citra yang memiliki rentang kontras lebih lebar, sehingga detail lebih terlihat.

**Menampilkan Citra Asli dan Citra Hasil Peregangan Kontras**

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title('Original Image')

plt.imshow(image, cmap='gray')

plt.subplot(1, 2, 2)

plt.title('Contrast Stretched Image')

plt.imshow(stretched\_image, cmap='gray')

plt.show()

**Penjelasan:**

 **plt.figure()**: Menentukan ukuran tampilan gambar.

 **plt.subplot()**: Mengatur tampilan menjadi 1 baris dan 2 kolom, dengan citra asli di sebelah kiri dan citra hasil peregangan kontras di sebelah kanan.

 **plt.title()**: Memberikan judul pada setiap citra.

 **plt.imshow()**: Menampilkan gambar dalam skala keabuan (cmap='gray').

 **plt.show()**: Menampilkan kedua gambar di jendela output.

**Menentukan Konstanta c untuk Transformasi Log**:

c = 255 / np.log(1 + np.max(image))

**Penjelasan:**

* np.max(image): Mencari nilai piksel tertinggi pada citra.
* np.log(1 + np.max(image)): Mengambil logaritma dari nilai piksel maksimum ditambah satu (untuk menghindari log(0), yang tidak terdefinisi).
* 255 / np.log(1 + np.max(image)): Menentukan konstanta c untuk memastikan hasil transformasi memiliki rentang intensitas yang memadai (dari 0 hingga 255 pada citra 8-bit).

**Menerapkan Log Transformation:**

log\_image = (c \* np.log(1 + image)).astype(np.uint8)

**Penjelasan:**

* np.log(1 + image): Mengambil logaritma dari nilai tiap piksel, yang bertujuan untuk memperluas perbedaan pada intensitas rendah (area gelap) dan memperkecil perbedaan pada intensitas tinggi (area terang).
* c \* np.log(1 + image): Mengalikan hasil logaritma dengan konstanta c agar hasil transformasi berada dalam rentang intensitas penuh (0-255).
* astype(np.uint8): Mengonversi hasil ke tipe uint8, yang sesuai untuk citra 8-bit (rentang nilai 0-255).

Hasilnya adalah citra di mana detail dalam area gelap lebih menonjol, sementara area terang tidak terlalu dominan.

**Menampilkan Citra Asli dan Citra Hasil Log Transformation**:

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title('Original Image')

plt.imshow(image, cmap='gray')

plt.subplot(1, 2, 2)

plt.title('Log Transformed Image')

plt.imshow(log\_image, cmap='gray')

plt.show()

**Penjelasan:**

* plt.figure(): Menentukan ukuran tampilan gambar.
* plt.subplot(): Mengatur tata letak tampilan menjadi 1 baris dan 2 kolom, dengan citra asli di sebelah kiri dan citra hasil transformasi log di sebelah kanan.
* plt.title(): Memberikan judul pada setiap citra.
* plt.imshow(): Menampilkan gambar dalam skala keabuan (cmap='gray').
* plt.show(): Menampilkan kedua gambar di jendela output.