

Risang Panggalih

24/550961/PPA/06961

Laporan Tugas Mata Kuliah Computer Vision Lanjut

“Image Enhancement”

1. Metodologi

A. Histogram Equalization

Histogram Equalization dilakukan untuk meningkatkan kontras dari gambar yang memiliki kontras rendah (low-contrast). Metodologi yang digunakan untuk penerapan histogram equalization pada tugas ini mencakup perhitungan histogram citra dan Cumulative Distribution Function (CDF). Kemudian menggunakan CDF yang sudah dinormalisasi sebagai Look-Up Table (LUT) untuk mengubah (remap) intensitas pixel. Citra input yang digunakan adalah citra low-contrast seperti berikut:



low-contrast.png

B. Box Blur dan Unsharp Masking

Kedua metode ini digunakan untuk mengatasi gambar yang terlalu tajam (over-sharp) dan terlalu kabur (over-blur). Citra yang terlalu tajam menerapkan metode blur yaitu **Box Blur** (smoothing filter sederhana). Selanjutnya, hasil citra yang sudah diberikan filter smoothing digunakan untuk mempertajam citra menggunakan algoritma **Unsharp Masking** yaitu dengan mengurangi citra blur (unsharp) dengan citra asli untuk menghasilkan map detail. Map detail ini kemudian ditambahkan kembali ke citra asli untuk meningkatkan/mempertajam edge dan detail. Citra input yang digunakan adalah citra sebagai berikut:



akmal.png

2. Implementasi dan Hasil

A. Helper Function

- `load_img(path)`: Digunakan untuk memuat citra lalu melakukan konversi citra menjadi grayscale ('L'). Kemudian citra direpresentasikan menjadi NumPy array `uint8` (format standar untuk 8-bit image processing).
- `to_pil(arr)`: Digunakan untuk mengubah NumPy array Kembali menjadi citra PIL. `np.clip(arr, 0, 255)` digunakan untuk memastikan value pixel tetap berada dalam range 0-255 sebelum diubah tipe datanya.
- `show(arr, title)`: Digunakan sebagai wrapper Matplotlib untuk menampilkan citra.

B. `histogram_equalization(img)`

- Pertama-tama, dilakukan pengecekan apakah input merupakan array 2D (citra grayscale) untuk mencegah error yang mungkin ditimbulkan jika citra yang digunakan berwarna.
- `np.bincount(img.reshape(-1), minlength=256)` digunakan untuk menghitung histogram integer data. `hist.cumsum()` digunakan untuk mengukur CDF.
- Selanjutnya, `return lut[img]` menerapkan indexing NumPy untuk mengaplikasikan Look-Up Table (LUT) ke seluruh pixel citra sekaligus.
- Algoritma utama untuk menghitung equalization yaitu: $H(v) = \text{round}((\text{cdf}(v) - \text{cdf_min}) / (M*N - \text{cdf_min}) * (L-1))$.

Berikut adalah citra output dari proses histogram equalization:



Hasil Histogram Equalization

Citra "Sebelum" terlihat lebih pudar dan detail tidak terlihat dengan jelas. Citra "Sesudah" memiliki distribusi intensitas pixel yang lebih luas sehingga menghasilkan kontras yang lebih baik di mana detail citra dapat terlihat jelas.

C. `box_blur(arr)` dan `unsharp(arr)`

- Fungsi `_movavg1d(a, k)` membantu menjalankan proses moving average 1D. `np.pad(..., mode='edge')` digunakan untuk menangani tepi/border citra selama konvolusi/convolution dengan mencegah dark edges yang disebabkan oleh zero-padding.

- **box_blur(arr, k)** digunakan untuk melakukan blur dengan mengimplementasikan separable filter. Fungsi ini mengaplikasikan blur 1D secara horizontal dan kemudian vertikal.
- **unsharp(arr, k, amount)** digunakan untuk mengimplementasikan fungsi unsharp masking yaitu: $\text{sharpened} = \text{original} + \text{amount} * (\text{original} - \text{blurred})$. Selanjutnya tipe data array dikonversi menjadi np.float32 untuk mencegah data loss dan mengatasi negative value yang mungkin muncul pada tahap pengurangan/substraksi. np.clip(out, 0, 255) digunakan untuk memastikan value pixel yang ditajamkan valid sebelum diubah kembali menjadi uint8.

Berikut adalah citra output dari proses box blur dan unsharp masking:



Hasil Box Blur dan Unsharp Masking

Citra "Sesudah Box Blur" terlihat lebih halus, buram, dan tidak detail dibandingkan citra aslinya. Citra "Sesudah Unsharp Masking" tampak lebih tajam, dengan tepi dan tekstur yang lebih terlihat tajam.