

NAMA : VIENA DWI PUTRI MAULINA

NIM : 312110469

KELAS : TI.21.C1

MATA KULIAH : PENGOLAHAN CITRA

UJIAN AKHIR SEMESTER

LAPORAN PROGRAM YANG MENAPILKAN CITRA ASLI, HISTOGRAM, NILAI MSE DAN PSNR DENGAN METODE MEDIAN FILTER

Link Github : https://github.com/vienamaulina/UAS_PengolahanCitra.git

Citra asli

Citra asli merujuk pada gambar atau foto yang belum mengalami manipulasi atau perubahan. Ini berarti bahwa citra asli adalah representasi yang akurat dan tidak mengandung modifikasi, penyuntingan, atau manipulasi dalam bentuk apa pun. Citra asli sering digunakan dalam konteks forensik, analisis gambar, dan keaslian dokumentasi visual.

Histogram

Histogram adalah representasi grafis dari distribusi intensitas piksel dalam sebuah gambar. Histogram memberikan informasi tentang seberapa sering nilai intensitas piksel tertentu muncul di seluruh gambar. Histogram sering kali digunakan dalam analisis dan manipulasi citra untuk memahami karakteristik dan properti gambar. Histogram dapat memberikan wawasan tentang kecerahan, kontras, distribusi warna, dan informasi lainnya terkait intensitas piksel dalam gambar. Secara umum, histogram ditempatkan pada sumbu horizontal dengan rentang nilai intensitas piksel dari 0 hingga 255 (dalam skala keabuan) atau dari 0 hingga 1 (dalam skala warna). Sedangkan sumbu vertikal menunjukkan jumlah piksel dalam gambar yang memiliki intensitas tertentu.

MSE

MSE (Mean Squared Error) adalah sebuah metrik yang digunakan untuk mengukur perbedaan atau kesalahan antara dua gambar, yaitu gambar asli dan gambar hasil pengolahan atau rekonstruksi. MSE digunakan untuk mengevaluasi sejauh mana gambar hasil rekonstruksi mendekati gambar asli. MSE menghitung rata-rata dari kuadrat selisih antara nilai intensitas piksel pada posisi yang sesuai di kedua gambar. Semakin kecil nilai MSE, semakin dekat gambar hasil rekonstruksi dengan gambar asli, dan semakin sedikit kesalahan atau perbedaan antara keduanya.

PSNR

PSNR (Peak Signal-to-Noise Ratio) adalah metrik yang digunakan untuk mengukur kualitas rekonstruksi gambar dalam pengolahan citra. PSNR menggambarkan perbandingan antara energi sinyal yang diinginkan (gambar asli) dengan energi kesalahan atau noise yang terdapat dalam representasi hasil rekonstruksi. PSNR diukur dalam skala logaritmik dan biasanya dinyatakan dalam desibel (dB). Semakin tinggi nilai PSNR, semakin baik kualitas rekonstruksi gambar.

Median filter

Median filter adalah sebuah metode pemrosesan citra yang digunakan untuk mengurangi noise atau derau pada gambar. Tujuan utama dari median filter adalah menghaluskan gambar dengan mempertahankan tepi atau detail penting sementara menghilangkan piksel-piksel yang dianggap sebagai noise. Cara kerja median filter adalah dengan meluncurkan jendela berukuran kecil (misalnya, jendela 3x3 atau 5x5) pada setiap piksel dalam gambar. Kemudian, nilai piksel-piksel yang tercakup dalam jendela tersebut diurutkan secara ascending (menaik) atau descending (menurun), dan nilai median dari piksel-piksel tersebut digunakan sebagai nilai piksel baru di tengah jendela. Dengan menggunakan median filter, nilai piksel yang terpengaruh oleh noise akan dihilangkan dan digantikan dengan nilai median yang lebih representatif dari area sekitarnya. Hal ini membantu dalam mengurangi efek derau seperti titik-titik berisolasi, puncak-puncak tajam, atau garis-garis yang tidak diinginkan dalam gambar.

Berikut programnya

```
import tkinter as tk
from tkinter import filedialog
from PIL import ImageTk, Image
import numpy as np
import cv2
import matplotlib.pyplot as plt

def calculate_mse(image1, image2):
    squared_diff = np.square(image1.astype("float") - image2.astype("float"))
    mse = np.mean(squared_diff)
    return mse

def calculate_psnr(mse, max_pixel=255.0):
    if mse == 0:
        return float("inf")
    psnr = 20 * np.log10(max_pixel / np.sqrt(mse))
```

```

    return psnr

def apply_median_filter(image):
    median_filtered = cv2.medianBlur(image, 3)
    return median_filtered

def open_image():
    global original_image, mse_label, psnr_label

    path = filedialog.askopenfilename(filetypes=[("Image files", "*.png *.jpg *.jpeg")])
    if path:
        image = Image.open(path)
        original_image = np.array(image)
        show_image_histogram(image)
        show_filtered_image()

        filtered_image = apply_median_filter(original_image)
        mse = calculate_mse(original_image, filtered_image)
        psnr = calculate_psnr(mse)

        mse_label.config(text="MSE: {:.2f}".format(mse))
        psnr_label.config(text="PSNR: {:.2f}".format(psnr))

def show_image_histogram(image):
    image = image.resize((300, 225), Image.ANTIALIAS)
    photo = ImageTk.PhotoImage(image)
    image_label.configure(image=photo)
    image_label.image = photo

    image_gray = cv2.cvtColor(np.array(image), cv2.COLOR_RGB2GRAY)
    histogram = cv2.calcHist([image_gray], [0], None, [256], [0, 256])
    plt.figure()
    plt.title('Histogram')
    plt.xlabel('Intensity')
    plt.ylabel('Count')
    plt.plot(histogram)
    plt.xlim([0, 256])
    plt.tight_layout()
    plt.savefig('histogram.png')
    plt.close()

```

```

    histogram_image = Image.open('histogram.png')
    histogram_image = histogram_image.resize((300, 225), Image.ANTIALIAS)
    histogram_photo = ImageTk.PhotoImage(histogram_image)
    histogram_label.configure(image=histogram_photo)
    histogram_label.image = histogram_photo

def show_filtered_image():
    filtered_image = apply_median_filter(original_image)
    filtered_image = Image.fromarray(filtered_image)
    filtered_image = filtered_image.resize((300, 225), Image.ANTIALIAS)
    filtered_photo = ImageTk.PhotoImage(filtered_image)
    filtered_image_label.configure(image=filtered_photo)
    filtered_image_label.image = filtered_photo

root = tk.Tk()
root.title("Image Processing")

# Create a grid layout
root.grid_columnconfigure(0, weight=1)
root.grid_rowconfigure(2, weight=1)

button_frame = tk.Frame(root)
button_frame.grid(row=0, column=0, pady=10)

open_button = tk.Button(button_frame, text="Open Image", command=open_image)
open_button.pack()

image_label = tk.Label(root)
image_label.grid(row=1, column=0, padx=10)

histogram_label = tk.Label(root)
histogram_label.grid(row=2, column=0, padx=10)

filtered_image_label = tk.Label(root)
filtered_image_label.grid(row=1, column=1, padx=10)

metrics_frame = tk.Frame(root)
metrics_frame.grid(row=2, column=1, pady=10)

mse_label = tk.Label(metrics_frame, text="MSE: -")
mse_label.pack()

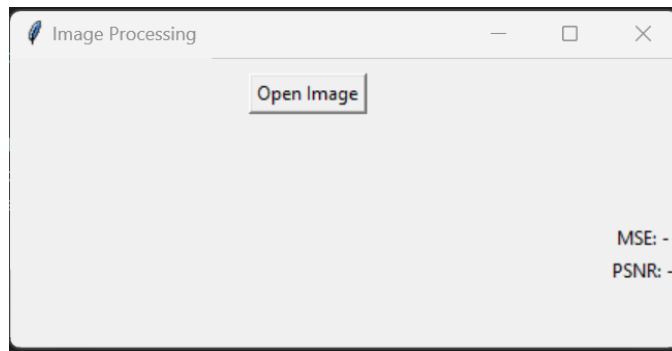
psnr_label = tk.Label(metrics_frame, text="PSNR: -")

```

```
psnr_label.pack()  
  
root.mainloop()
```

Berikut Outputnya

Klik open > pilih foto



Maka hasilnya seperti ini

