Modul Praktikum: Dasar Pengolahan Citra dengan OpenCV

Mata Kuliah: Pengolahan Citra Digital

Disusun Oleh: Departemen Teknik Informatika

Deskripsi Modul

Modul praktikum ini dirancang untuk memperkenalkan dasar-dasar pengolahan citra digital menggunakan library OpenCV dalam bahasa pemrograman Python. Praktikum ini mencakup operasi dasar seperti membaca, menampilkan, dan menyimpan gambar, serta manipulasi warna, transformasi geometris, dan operasi filtering.

Prasyarat

- Pengetahuan dasar Python
- Library OpenCV, NumPy, dan Matplotlib terinstall
- Akses ke gambar contoh untuk praktikum

Instalasi

```
!pip install opencv-python
!pip install numpy
!pip install matplotlib
```

Materi Praktikum

1. Import Library

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

2. Membaca dan Menampilkan Gambar

```
# Membaca gambar
img = cv2.imread('gambar.jpg')

# Menampilkan gambar dengan OpenCV
cv2.imshow('Gambar Asli', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Menampilkan gambar dengan Matplotlib
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Gambar Asli')
plt.axis('off')
plt.show()
```

3. Mengakses Properti Gambar

```
# Mendapatkan dimensi gambar
tinggi, lebar, channel = img.shape
print(f"Dimensi gambar: {lebar} x {tinggi}")
print(f"Jumlah channel: {channel}")

# Mengakses nilai piksel
piksel = img[100, 100]  # Koordinat (y, x)
print(f"Nilai piksel di (100,100): {piksel}")
```

4. Manipulasi Warna

```
# Konversi warna
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Menampilkan hasil konversi
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_gray, cmap='gray')
plt.title('Grayscale')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(img_hsv)
plt.title('HSV')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(img_rgb)
plt.title('RGB')
plt.axis('off')

plt.show()
```

5. Transformasi Geometris

```
# Resize gambar
img_resized = cv2.resize(img, (300, 200))

# Rotasi gambar
tinggi, lebar = img.shape[:2]
pusat = (lebar // 2, tinggi // 2)
matriks_rotasi = cv2.getRotationMatrix2D(pusat, 45, 1.0)  # Rotasi 45 derajat
img_rotated = cv2.warpAffine(img, matriks_rotasi, (lebar, tinggi))

# Transformasi affine
pts1 = np.float32([[50, 50], [200, 50], [50, 200]])
pts2 = np.float32([[10, 100], [200, 50], [100, 250]])
matriks_affine = cv2.getAffineTransform(pts1, pts2)
img_affine = cv2.warpAffine(img, matriks_affine, (lebar, tinggi))

# Menampilkan hasil transformasi
plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Gambar Asli')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB))
plt.title('Resized')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(cv2.cvtColor(img_rotated, cv2.COLOR_BGR2RGB))
plt.title('Rotated')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(img_affine, cv2.COLOR_BGR2RGB))
plt.title('Affine Transform')
plt.axis('off')

plt.show()
```

6. Operasi Filtering

```
 # Blurring
img_blur = cv2.blur(img, (5, 5))
img_gaussian = cv2.GaussianBlur(img, (5, 5), 0)
img_median = cv2.medianBlur(img, 5)

# Sharpening
kernel_sharpen = np.array([[-1, -1, -1],
                           [-1, 9, -1],
                           [-1, -1, -1]])
img_sharp = cv2.filter2D(img, -1, kernel_sharpen)

# Menampilkan hasil filtering
plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)
plt.imshow(cv2.cvtColor(img_blur, cv2.COLOR_BGR2RGB))
plt.title('Average Blur')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(cv2.cvtColor(img_gaussian, cv2.COLOR_BGR2RGB))
plt.title('Gaussian Blur')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(cv2.cvtColor(img_median, cv2.COLOR_BGR2RGB))
plt.title('Median Blur')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(img_sharp, cv2.COLOR_BGR2RGB))
plt.title('Sharpened')
plt.axis('off')

plt.show()
```

7. Thresholding

```
 # Konversi ke grayscale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Binary threshold
_, thresh_binary = cv2.threshold(img_gray, 127, 255, cv2.THRESH_BINARY)

# Adaptive threshold
thresh_adaptive = cv2.adaptiveThreshold(img_gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                        cv2.THRESH_BINARY, 11, 2)

# Otsu's threshold
_, thresh_otsu = cv2.threshold(img_gray, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# Menampilkan hasil thresholding
plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)
plt.imshow(img_gray, cmap='gray')
plt.title('Grayscale')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(thresh_binary, cmap='gray')
plt.title('Binary Threshold')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(thresh_adaptive, cmap='gray')
plt.title('Adaptive Threshold')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(thresh_otsu, cmap='gray')
plt.title("Otsu's Threshold")
plt.axis('off')

plt.show()
```

8. Operasi Morfologi

```
 # Kernel untuk operasi morfologi
kernel = np.ones((5, 5), np.uint8)

# Erosion
img_erosion = cv2.erode(thresh_binary, kernel, iterations=1)

# Dilation
img_dilation = cv2.dilate(thresh_binary, kernel, iterations=1)

# Opening (erosion followed by dilation)
img_opening = cv2.morphologyEx(thresh_binary, cv2.MORPH_OPEN, kernel)

# Closing (dilation followed by erosion)
img_closing = cv2.morphologyEx(thresh_binary, cv2.MORPH_CLOSE, kernel)

# Menampilkan hasil operasi morfologi
plt.figure(figsize=(15, 10))

plt.subplot(2, 3, 1)
plt.imshow(thresh_binary, cmap='gray')
plt.title('Original Binary')
plt.axis('off')

plt.subplot(2, 3, 2)
plt.imshow(img_erosion, cmap='gray')
plt.title('Erosion')
plt.axis('off')

plt.subplot(2, 3, 3)
plt.imshow(img_dilation, cmap='gray')
plt.title('Dilation')
plt.axis('off')

plt.subplot(2, 3, 4)
plt.imshow(img_opening, cmap='gray')
plt.title('Opening')
plt.axis('off')

plt.subplot(2, 3, 5)
plt.imshow(img_closing, cmap='gray')
plt.title('Closing')
plt.axis('off')

plt.show()
```

Tugas Praktikum

1. Lakukan semua operasi dasar pada gambar pilihan Anda
2. Eksperimen dengan parameter yang berbeda pada setiap operasi
3. Bandingkan hasil dari berbagai metode thresholding dan filtering
4. Buat laporan singkat tentang hasil eksperimen Anda

Referensi

- OpenCV Documentation: https://docs.opencv.org/
- NumPy Documentation: https://numpy.org/doc/
- Matplotlib Documentation: https://matplotlib.org/stable/contents.html
- Colab Notebook: https://colab.research.google.com/github/danylaksono/OpenCV-PCD/blob/main/Minggu%202_%20Dasar%20Pengolahan%20Citra%20dengan%20OpenCV.ipynb