CMA 103/ Topik Dalam Image Processing

Ujian Tengah Semester

Muhammad Fajrul Aslim – 20200804031

---

1. [20%] Berikan definisi atau pengertian anda mengenai image berikut ini:

JAWAB

Gambar di atas disebut *color image* (gambar berwarna) karena berisi informasi warna. *Color image* terdiri dari 3 *grayscale image* yang masing-masingnya memiliki warna yang berbeda seperti warna merah (*red*), hijau (*green*), dan biru (*blue*). Dengan kata lain *color image* dapat diperoleh dari kamera yang memiliki 3 sensor warna.

Untuk memproses *color image*, juga dibutuhkan waktu komputasi yang lebih lama dibandingkan untuk memproses *grayscale image* ataupun *black and white image*. Hal ini disebabkan memproses *color image* sama seperti memproses 3 buah *grayscale image* ataupun *black and white image*.

2. [10%] Jelaskan persamaan dan perbedaan sistem visual manusia kamera digital

JAWAB

Mata manusia dan kamera digital sama-sama memiliki focus terhadap sebuah benda. Keduanya sama-sama mampu menyesuaikan jumlah cahaya yang masuk.

Korne, atau selaput mata bagian luar, ibarat lensa dari kamera. Iris dan pupil berfungsi seperti lubang cahaya di kamera. Lalu retina mirip sensor gambar pada kamera digital.

Tapi perbedaan yang mendasar adalah pada focus lensanya dan sensitivitas terhadap datangnya cahaya. Tapi, yang pasti, kameralah yang meniru fungsi mata dan bukan sebaliknya.

3. [20%] Jelaskan persamaan dan perbedaan pengolahan gambar pada domain spasial dan domain frekuensi. Berikan contoh.
   JAWAB

Domain Spasial	**Domain Frekuensi**
Dasar untuk filter linear adalah teori konvolusi.	Dasar untuk filter linear adalah teori konvolusi.
Konsep koordinat baris dan kolom	Konsep frekuensi, perubahan intensitas piksel ke piksel (frekuensi rendah ke tinggi)
Pemrosesan piksel per piksel	Pemrosesan berdasarkan pemilihan frekuensi yang akan difilter atau tidak
Komputasi lama (terutama citra dengan ukuran spasial tinggi)	Komputasi relative cepat (terutama citra dengan ukuran spasial tinggi
Teknik pemrosesan pada domain spasial didasarkan pada manipulasi piksel dalam citra secara langsung.	Teknik pemrosesan pada domain frekuensi didasarkan pada manipulasi terhadap transformasi Fourier dari suatu citra.
Contoh: Peningkatan kualitas gambar dengan *image enhancement*. Manipulasi dilakukan terhadap piksel di dalam image sesuai koordinatnya.	**Contoh:** Peningkatan kualitas gambar dengan *image enhancement*. Manipulasi dilakukan dengan mengubah *image* ke dalam ruang frekuensi dengan transformasi Fourier terlebih dahulu
Contoh: Teknik penyisipan *watermark* dalam sebuah citra. Penyisipan watermark dilakukan dengan melakukan pengubahan bit-bit data secara langsung pada data spasial citra penampungnya. Contohnya adalah penyisipan watermark pada LSB (Least Significant Bit).	**Contoh:** Teknik penyisipan *watermark* dalam sebuah citra. Penyisipan watermark dilakukan dengan cara melakukan transformasi pada data penampung, kemudian perubahan dilakukan terhadap koefisien transformasinya. Contohnya adalah penyisipan watermark di ranah frekuensi dengan terlebih dahulu melakukan transformasi DCT (Discrete Cosine Transform).

4. Image Enhancement. [20%] Teknik operasi poin manakah dapat anda gunakan untuk meningkatkan kualitas image tersebut (lihat slide pada sesi 4). Jelaskan prinsip kerja operasi yang anda pilih tersebut
   JAWAB
   Teknik operasi poin *image averaging*, dengan prinsip kerja sebagai berikut:

1) Misalkan citra bernoise g(x,y) dibentuk dengan menambahkan noise $\eta(x,y)$ ke citra asal f(x,y), dengan rumus berikut:
   $g(x,y)=f(x,y)+\eta(x,y)$
   Diasumsikan bahwa pada setiap pasangan koordinat (x,y), noise tidak berkorelasi dan memiliki nilai rata-rata 0.
2) Tujuan dari "image averaging" adalah mengurangi noise dengan cara merata-rata sekumpulan citra bernoise, $\{g_i(x,y)\}$.
3) Jika noise memenuhi batasan seperti yang disebutkan sebelumnya, bisa ditunjukkan bahwa jika suatu citra gg(x,y) dibentuk dengan merata-rata K buah citra bernoise
4) Ketika K bertambah, variasi noise pada nilai-nilai piksel di setiap lokasi (x,y) menurun.
5) Karena E{gg(x,y)} = f(x,y), maka gg(x,y) akan mendekati f(x,y) ketika jumlah citra bernoise yang dirata-rata bertambah.

5. Filtering. [20%] Jelaskan filter yang akan anda pilih untuk meningkatkan kualitas image tersebut. Jelaskan proses konvolusi dari penerapan filter pada image tersebut.
   JAWAB
   Metode filtering yang menggunakan proses kovolusi pada citra menggunakan Metode Sharpen.
   Sharpening (Penajaman) yaitu memperjelas detil suatu citra(menambah kontras) dengan penjumlahan atas citra tepi dengan citra aslinya maka bagian tepi objek akan terlihat berbeda dengan latarnya, sehingga citra terkesan lebih tajam.
   Proses penajaman berhubungan dengan deteksi tepi - perubahan warna yang dilemahkan untuk menciptakan efek tepi tajam. Menggunakan fspecial yang akan membuat filter untuk menajamkan (sharpening) gambar. Filter khususnya bernama 'unsharp'.

   Penjelasan Input-Proses-Output Metode Sharpen
   1) Tahap Input Citra
      Pada tahap ini merupakan awal proses penelitian dengan melakukan pengambilan citra. Ada beberapa file citra yang didukung oleh Matlab, yaitu citra dengan format bitmap (*.bmp), JPEG (*.jpg), png (*.png) dan tif (*.tif). Citra original dan informasi citra ditampilkan beserta histogramnya.
   2) Tahap Proses Sharpening
      Tahap ini merupakan tindak lanjut dari tahap bluring pada citra dimana citra yang sudah mengalami proses blur diproses dengan filter laplaciandengan mask 3x3, dimana

0	1	0
1	-4	1
0	1	0

Mask 1

1	1	1
1	-8	1
1	1	1

Mask 2 Filter Sobel

-1	-2	-1
0	0	0
1	2	1

Mask 1

-1	0	1
2	0	2
-1	0	1

Proses perhitungan

Proses penajaman citra adalah dengan melakukan pengurangan smoothed dari citra asli (unsharp masking) proses ini terdiri dari langkah-langkah berikut :

a. Proses Degradasi Gambar Asli:
   Mengaburkan gambar (blurred image) dengan memberikan efek blur pada Matlab
b. Proses Restorasi (Enhancement):
   Mengurangi citra yang kabur, yaitu proses konvolusi menggunakan filter laplacian dan filter gradien.
   Tambahkan mask ke citra, citra hasil konvolusi ditambahkan ke citra blur.

SOAL PENERAPAN

1. [10%] Terapkan suatu Teknik operasi poin atau teknik filter pada suatu gambar menggunakan (Matlab/Python/dll) Jelaskan kodenya dan hasilnya.
   Petunjuk: carilah tutorial di Internet, tetap berikan jawaban kode dan jelaskan walaupun sekiranya hasilnya error
   JAWAB
   Python OpenCV: Membuat Filter Gambar Mirip Instagram
   Pada bagian ini, saya akan menjelaskan logika di balik bagaimana gambar RGB disimpan dan dimanipulasi oleh komputer. Jika Anda sudah terbiasa dengan hal ini, silakan lompat ke bagian selanjutnya di mana kita akan melompat ke detail yang lebih lanjut.
   Data unit dasar dalam gambar adalah piksel . Piksel hanyalah satu titik dalam gambar dan disimpan sebagai angka dalam kisaran [0, 256]. Model RGB adalah singkatan dari Red-Green-Blue dan memberitahu kita bahwa untuk setiap piksel kita menyimpan

intensitas merah, hijau dan biru (kita akan menyebut saluran ini) sebagai angka dari 0 hingga 256. Pixel berwarna putih jika semua 3 saluran adalah 256 dan hitam jika semua 3 saluran adalah 0. Sebuah piksel sepenuhnya merah / hijau / biru jika saluran masing-masing adalah 256 dan semua saluran lainnya adalah 0. Anda mengerti, setiap warna akan direpresentasikan sebagai campuran dari 3 saluran ini.
Jadi gambar adalah kumpulan piksel. Jika, katakanlah, gambar kita 300x200, maka kita akan menyimpannya sebagai larik 2D dengan 300 garis dan 200 kolom di mana setiap sel adalah piksel. Tapi kita tahu dari atas bahwa untuk setiap piksel kita akan menyimpan informasi tentang semua 3 saluran, jadi itu sebenarnya memberi kita larik 3D.

PENYIAPAN PROJEK
Kita perlu menginstal 2 paket python dan kemudian kita siap melakukannya.

```
pip3 install opencv-python
pip3 install scipy
```

```
initialImage = cv2.imread("image1.jpg")
blurredImage = gaussianBlur(copy.deepcopy(initialImage))
cv2.imwrite("blurred.jpg", blurredImage)
```

Ini adalah bagian di mana kita akan menerapkan konvolusi dengan sekumpulan kernel yang telah ditentukan untuk mendapatkan efek yang indah. Untuk jenis transformasi lainnya, silakan lompat ke bagian berikutnya. Untuk setiap transformasi, saya akan menunjukkan kernel dan kodenya dan di akhir bagian ini, saya akan menampilkan galeri dengan gambar awal dan hasilnya.

PENAJAMAN GAMBAR

-1	-1	-1
-1	9	1
-1	-1	-1

Ini adalah kernel yang digunakan untuk mempertajam detail pada gambar. Kami akan menggunakan metode filter2D dari perpustakaan OpenCV yang akan melakukan konvolusi untuk kami.

```
def sharpen(image):
    kernel = np.array([[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]])
```

```
    return cv2.filter2D(image, -1, kernel)
```

0.272	0.534	0.131
0.349	0.686	0.168
0.393	0.769	0.189

```
def sepia(image):
    kernel = np.array([[0.272, 0.534, 0.131],
                       [0.349, 0.686, 0.168],
                       [0.393, 0.769, 0.189]])
    return cv2.filter2D(image, -1, kernel)
```

Untuk efek ini kita bisa menggunakan kernel dasar seperti semua hal di atas, tetapi hasilnya cukup timpang. Untungnya, OpenCV telah menerapkan gaussian blur yang akan melakukan pekerjaan itu untuk kita. Yang perlu kita lakukan adalah:

```
def gaussianBlur(image):
    return cv2.GaussianBlur(image, (35, 35), 0)
```

0	-1	-1
1	0	-1
1	1	0

```
def emboss(image):
    kernel = np.array([[0,-1,-1],
                           [1,0,-1],
                           [1,1,0]])
    return cv2.filter2D(image, -1, kernel)
```

Gambar awal

Dari kiri ke kanan: Buram, Timbul, Sepia, Tajam

OpenCV - transformasi tabel pencarian

Kita akan menggunakan jenis transformasi ini untuk membuat gambar tampak lebih hangat dan lebih dingin, tetapi pertama-tama mari kita lihat bagaimana kita dapat menggunakan tabel pencarian untuk itu.

Sebuah tabel adalah hanya kumpulan sederhana dari pasangan nilai tampak seperti ini: [. Value1, value2, ..., valueN], [modifiedValue1, modifiedValue2, ..., modifiedValueN] dan logika di balik ini adalah sederhana - kita akan mengambil setiap pixel nilai dan menggantinya dengan nilai yang sesuai dari tabel pencarian (artinya ganti nilai1 dengan modifiedValue1 dan seterusnya).

Masalahnya, ada banyak nilai yang harus diganti (dari 0 hingga 256) dan membuat tabel pencarian untuk ini adalah proses yang menyakitkan. Tapi keberuntungan ada di pihak kita lagi, karena ada alat yang bisa kita gunakan untuk itu.

The UnivariateSpline smoothing metode dari scipy paket di sini untuk membantu kami. Metode ini hanya membutuhkan sedikit nilai referensi dan mencoba menemukan metode untuk mengubah semua nilai lain dalam rentang yang terkait dengan nilai referensi yang telah kami sediakan.

Pada dasarnya, kita akan mendefinisikan tabel pencarian singkat seperti ini
[0, 64, 128, 256], [0, 80, 160, 256]
dan terapkan transformasi UnivariateSpline yang akan mengisi sisa nilai dalam rentang [0, 256].

Setelah tabel pencarian kami dibuat, kami hanya perlu menerapkannya ke saluran tertentu. Untuk mendapatkan gambar hangat , kita akan meningkatkan nilai saluran merah dan menurunkan nilai saluran biru untuk semua piksel dalam gambar. Untuk mendapatkan gambar dingin , kita akan melakukan yang sebaliknya: meningkatkan nilai untuk saluran biru dan menurunkan nilai untuk saluran merah. Saluran hijau menghapus tidak tersentuh dalam kedua kasus.

```
def spreadLookupTable(x, y):
  spline = UnivariateSpline(x, y)
  return spline(range(256))
def warmImage(image):
```

```
    increaseLookupTable = spreadLookupTable([0, 64, 128, 256], [0, 80, 160, 256]
)
    decreaseLookupTable = spreadLookupTable([0, 64, 128, 256], [0, 50, 100, 256]
)
    red_channel, green_channel, blue_channel = cv2.split(image)
    red_channel = cv2.LUT(red_channel, increaseLookupTable).astype(np.uint8)
    blue_channel = cv2.LUT(blue_channel, decreaseLookupTable).astype(np.uint8)
    return cv2.merge((red_channel, green_channel, blue_channel))
def coldImage(image):
    increaseLookupTable = spreadLookupTable([0, 64, 128, 256], [0, 80, 160, 256]
)
    decreaseLookupTable = spreadLookupTable([0, 64, 128, 256], [0, 50, 100, 256]
)
    red_channel, green_channel, blue_channel = cv2.split(image)
    red_channel = cv2.LUT(red_channel, decreaseLookupTable).astype(np.uint8)
    blue_channel = cv2.LUT(blue_channel, increaseLookupTable).astype(np.uint8)
    return cv2.merge((red_channel, green_channel, blue_channel))
```

Gambar kiri hangat, Gambar tengah asli, Gambar kanan dingin