

PRAKTIKUM 2

Pengenalan citra digital

Materi

- Tipe, Format dan Ukuran Image
- Read dan Write Image
- Konversi Image

Tujuan Praktikum

- Mahasiswa dapat menyimpan *image* baru dengan format tertentu.
- Mahasiswa dapat menjalankan fungsi-fungsi dasar OpenCV untuk *write* , *convert*

A. Penyajian

Terdapat 4 tipe dasar dari sebuah image:

1. *Binary*

Piksel citra hanya terdiri dari warna hitam dan putih. Karena hanya ada dua nilai kemungkinan untuk setiap piksel, sehingga hanya perlu satu bit per piksel. Citra biner sangat efisien dalam hal penyimpanan.



Gambar 2.1. *Binary Image*

2. Grayscale

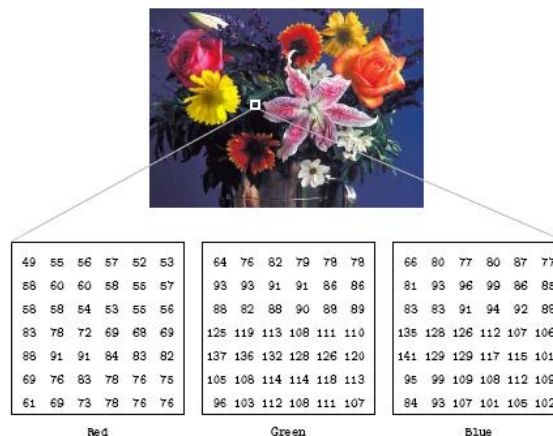
Setiap piksel adalah warna abu-abu, biasanya dari 0 (hitam) sampai 255 (putih). Setiap piksel direpresentasikan oleh 8 bits, atau 1 byte.



Gambar 2.2. *Grayscale Image*

3. True Color / RGB (*Red Green Blue*)

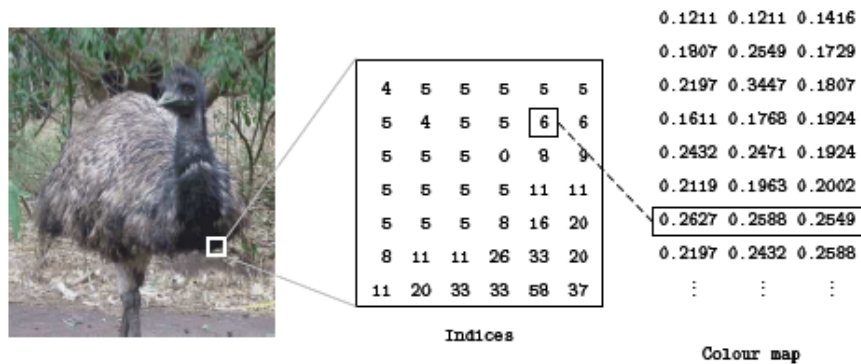
Pada tipe citra ini, setiap piksel mempunyai warna tertentu yaitu jumlah warna merah, hijau dan biru di dalamnya. Masing-masing komponen warna berkisaran dari 0-255, maka total komponen warna untuk setiap citra $255^3 = 16.777.216$ kemungkinan perbedaan citra dalam sebuah image. Setiap warna direpresentasikan ke dalam 8 *bits* sehingga RGB Image disebut juga *24-bit colour images*.



Gambar 2.3. *RGB Image*

4. *Indexed*

Kebanyakan citra hanya memiliki sebagian kecil dari lebih enam belas juta kemungkinan warna. Untuk kenyamanan penyimpanan dan penanganan file, citra memiliki sebuah *associated colour map* atau *colour palette*, yang merupakan daftar semua warna yang digunakan dalam sebuah citra. Setiap piksel memiliki nilai yang tidak memberikan warna (seperti pada citra RGB), tetapi indeks untuk warna dalam *map*.



Gambar 2.4. *Indexed colour image*

Berikut cara mengakses pixel dalam OpenCV:

```
// Akses 1 channel
Scalar intensity = img.at<uchar>(x, y);

// Akses 3 channel
Vec3b intensity = img.at<Vec3b>(x, y);
uchar blue = intensity.val[0];
uchar green = intensity.val[1];
uchar red = intensity.val[2];
```

Variable intensity merupakan intensitas pada pixel (x, y). Biasanya, *image* dalam bentuk digital berupa *true color* atau RGB. Namun untuk kepentingan tertentu, image perlu diubah ke dalam bentuk grayscale atau binary.

Berikut adalah formula yang digunakan untuk mengubah dari RGB ke Grayscale:

$$\text{Gray} = (\text{red} * 0.299 + \text{green} * 0.587 + \text{blue} * 0.114)$$

Untuk mendapatkan citra biner, biasanya menggunakan threshold pada citra grayscale, nilai threshold digunakan untuk menentukan apakah suatu pixel bernilai 0 atau 255.

Terdapat beberapa format citra:

1. BMP (Microsoft Windows Bitmap)
2. GIF (Graphics Interchange Files)

3. JPEG (Joint Photographic Experts Group)
4. PNG (Portable Network Graphics)
5. TIFF (Tagged Image File Format)

Pengubahan format citra pada OpenCV dapat dilakukan menggunakan fungsi `imwrite` yang akan menyimpan citra pada direktori tertentu.

Fungsi OpenCV

Terdapat beberapa fungsi OpenCV yang telah kita gunakan, antara lain:

1. Fungsi `imread`
→ Fungsi untuk membaca citra.

```
Mat imread
(
    const string& filename,
    int flags=1
)
```

2. Fungsi `cvtColor`
→ Fungsi untuk melakukan konversi citra

```
void cvtColor
(
    InputArray src,
    OutputArray dst,
    int code,
    int dstCn=0
)
```

3. Fungsi `imwrite`
→ Fungsi untuk menyimpan citra baru.

```
bool imwrite
(
    const string& filename,
    InputArray img,
    const vector<int>& params=vector<int>()
)
```

4. Fungsi `threshold`
→ Fungsi untuk memberikan threshold tertentu pada sebuah citra dengan menggunakan tipe threshold yang ada.

```
double threshold
(
    InputArray src,
    OutputArray dst,
    double thresh,
```

```

    double maxval,
    int type
)

```

Beberapa Tipe Threshold

Threshold Biner

$$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Threshold Biner Invers

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$$

Threshold Truncated

$$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

Threshold ToZero

$$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Threshold ToZero Inverse

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

B. LATIHAN

1. Membuat program untuk mengubah citra RGB menjadi citra grayscale kemudian disimpan ke dalam file baru dengan nama “output_citra.jpg” dengan menggunakan *library* OpenCV pada proyek “ImageLoad” yang telah dikonfigurasi.

Grayscale.cpp

```
#include <iostream>
#include <cv.h>
#include <highgui.h>

using namespace std;
using namespace cv;

int main()
{
    //Membaca citra bernama citra.tif
    Mat image1;
    image1 = imread("Lenna.jpg");
    Mat imGray = image1.clone();

    //Konversi warna dari RGB ke Grayscale
    cvtColor(image1, imGray, CV_RGB2GRAY);

    //Menyimpan citra
    imwrite("output_citra.jpg",imGray);

    return 0;
}
```

2. Membuat dan menampilkan citra biner

Biner.cpp

```
#include <iostream>
#include <cv.h>
#include <highgui.h>

using namespace std;
using namespace cv;
```

```

int main()
{
    //Membaca citra bernama citra.tif
    Mat image1,dst;
    image1 = imread("Lenna.jpg");
    Mat imGray = image1.clone();

    //Konversi warna dari RGB ke Grayscale
    cvtColor(image1, imGray, CV_RGB2GRAY);

    //Fungsi Thresholding
    threshold(imGray,dst,100,255,1);

    imshow("threshold",dst);
    cvWaitKey(0);

    return 0;
}

```

C. LKP

Buatlah proram untuk mengolah citra Lenna.jpg berdasarkan ketentuan berikut

- Ubah citra Lenna.jpg dari true color menjadi grayscale
- Ubah citra grayscale tersebut menjadi citra biner
- Ubah citra grayscale menggunakan threshold truncated

Semua program dibuat tanpa fungsi opencv kecuali fungsi untuk membaca, menampilkan, atau menulis citra.