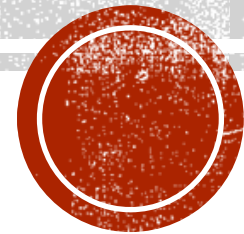


M11-IMAGE RETRIEVAL KNOWLEDGE DISCOVERY

Nama : Ni Putu Devira Ayu Martini
Kelas : S2 Teknik Elektro 2020
NRP : 1120800012



Assignment

- Provided general purpose image database containing of 1000 JPEG from COREL image collections
- These images are manually divided into 10 categories which are people, beaches, historian buildings, buses, dinosaurs, elephants, roses, horses, mountains, and foods



Assignment

- Build an image retrieval system for the dataset
- Use LOO for the testing
- Determine 15 top correct retrieved images for the query
- Analyze the performance for each image category with:
 - Error Ratio

$$Error = \sum_{i=1}^{15} err_i \begin{cases} err_i = 0 \leftarrow cr_i = cq \\ err_i = 1 \leftarrow otherwise \end{cases}$$

- Scoring

$$Score = \sum_{i=1}^{15} scr_i \begin{cases} scr_i = 15 - i + 1 \leftarrow cr_i = cq \\ scr_i = 0 \leftarrow otherwise \end{cases}$$



PERALATAN

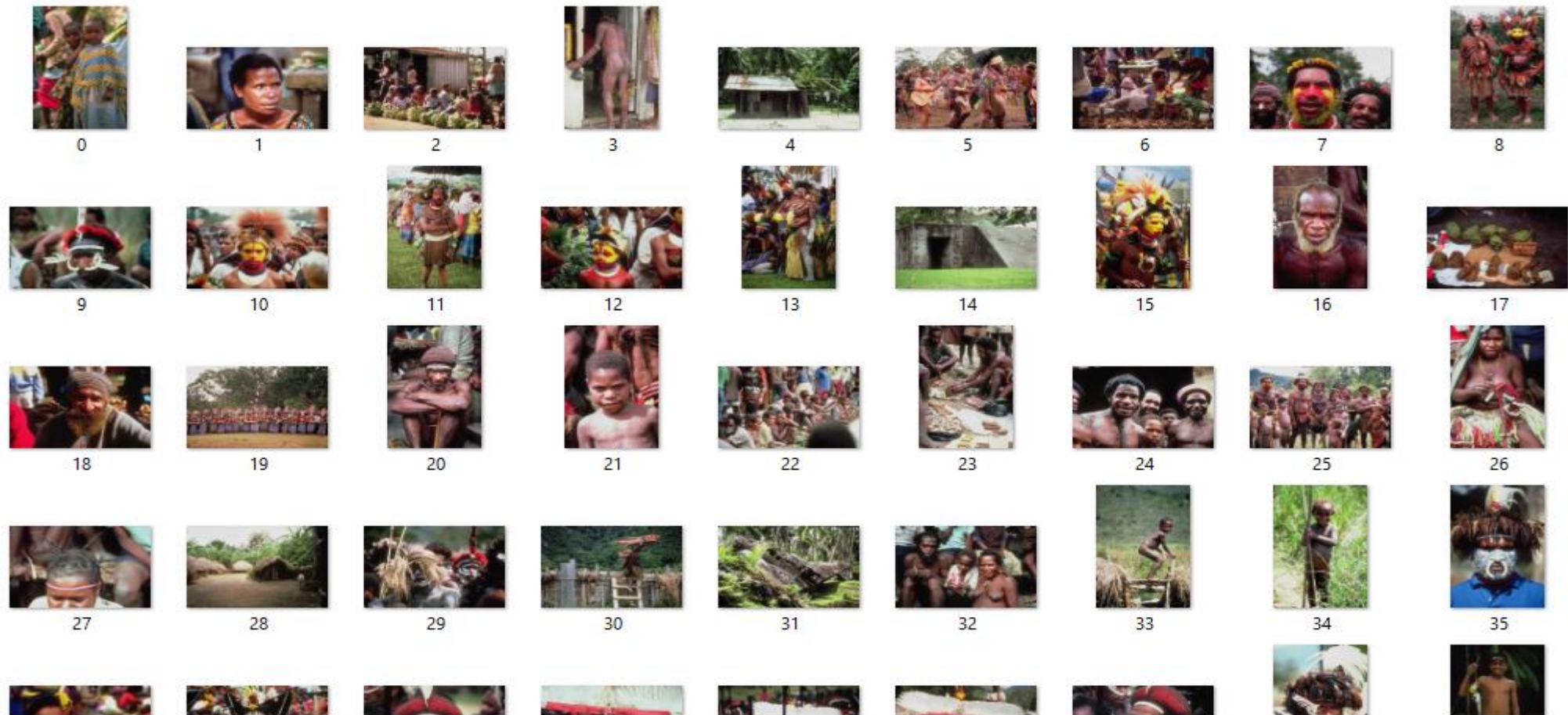
1. PC
2. Software Apache NetBeans
3. Dataset Corel



LANGKAH PERCOBAAN



MENYIMPAN DATASET GAMBAR 0-999 KE DALAM DIRECTORY



MENGIMPORT LIBRARY

```
package imageretrievalassignment;  
import ALI.*;
```

Memasukkan library ALI



INISIALISASI DATA GAMBAR

```
public class ImageRetrievalAssignment {  
    public static void main(String[] args) {  
        VectorLib vlib = new VectorLib();  
        ImageLib imgsearch = new ImageLib();
```

```
        String srcdir="C:\\Users\\user\\Desktop\\corel_dataset\\Image DB/";  
        String destdir="C:\\Users\\user\\Desktop\\corel_dataset\\Color DB/";  
        String cvqdir="C:\\Users\\user\\Desktop\\corel_dataset\\cvq DB/";  
        String srcfile, destfile;
```

Menginputkan
directory data
gambar

Menyimpan hasil
data Color dan cvq
ke directory



EKSTRAKSI FITUR WARNA MENGGUNAKAN CVQ

```
int[][][]rgb_colors=null;
double[][]cvq=new double[1000][1000];
double[][]images=new double[1000][1000];

for(int i=0; i<1000; i++)
{
    srcfile=srcdir+String.valueOf(i)+".jpg";
    rgb_colors=imgsearch.getRGB(srcfile);
    cvq[i]=imgsearch.ColorFeatureExtraction(rgb_colors);
}
vlib.saveDST(cvq,"cvq");
```

Array [1000] pertama =
banyaknya data pada directory
Array [1000] kedua = Banyaknya
setiap kategori, yaitu 10 kategori.
Maka, $10^3 = 1000$ bins

Memanggil dan memberi nama
setiap gambar, lalu kemudian
dilakukan ekstraksi warna.



KONVERSI GAMBAR & MEMUAT MATADATA WARNA

```
for(int i=0; i<1000; i++)  
{  
    srcfile=srcdir+String.valueOf(i)+".jpg";  
    destfile=cvqdir+String.valueOf(i)+".jpg";  
    imgsearch.RGB_to_CVQ_Image(srcfile, destfile, "jpg", 1000);  
}
```


```
//Loading color Features  
cvq=vlib.readDST("cvq");  
for(int i=0; i<1000; i++)  
{  
    destfile=destdir+"cvq"+String.valueOf(i);  
    images[i]=vlib.copyArray(cvq[i]);  
}
```

Memanggil data cvq yang
pada slide sebelumnya sudah
disimpan



MELAKUKAN LOOPING TESTING DATA GAMBAR MENGGUNAKAN TEKNIK LOO

```
int loop;  
for(loop=0; loop<=999; loop++)  
{  
    ....  
    ....  
}
```



Testing Data gambar menggunakan teknik LOO (Leave-One-Out) yang artinya menguji satu persatu dataset. Ada 1000 dataset "Corel", sehingga melakukan looping dari data ke-0 hingga 999.



MEMBERIKAN QUERY & MENENTUKAN KEMIRIPAN GAMBAR

```
int images_ke = loop;  
double[] query=vlib.copyArray(images[images_ke]);  
double[][] hasil=imgsearch.SimilarityMeasurement("cosine", query, images);  
vlib.view(hasil);
```

```
String categories = "None";
```

```
int[] Score = {0,0};
```

```
int Score_persent;
```

```
int[] Error = {0,0};
```

```
int TotalError;
```

Inisialisasi Variabel

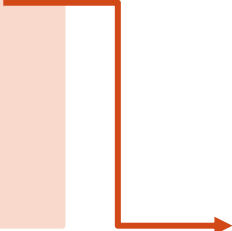
Memberikan query yaitu
“loop”, yang nantinya
“loop” tersebut berisi nama
dari gambar dataset.

Menentukan kemiripan
gambar 1 dengan gambar
yang lainnya, dengan
perhitungan jarak “cosine”,
Adapun tersedia perhitungan
lain yaitu “Euclidean,
Minkowski, dll”



MELAKUKAN PERANGKINGAN DARI DATA JARAK KEMIRIPAN

```
System.out.print("\n Top 15 Images: ");  
    for (int i=0; i<15; i++)  
    {  
        System.out.print("\n"+hasil[1][i]+",");  
        ....  
        ....  
    }
```



Melakukan perangkingan kemiripan gambar terbesar dari data gambar testing dengan dataset 0-999. Diambil hanya peringkat 0-14 atau 15 besar.



MELAKUKAN PROSES PERHITUNGAN SCORING & ERROR

```
if (images_ke >=0 && images_ke <= 99)
{
    categories = "People";
    if (hasil[1][i] >= 0.0 && hasil[1][i] <= 99.0)
    {
        Score[1]=Score[1]+15-i+1;
        Error[1]=Error[1]+1;
        System.out.print("\n Score[1]= "+Score[1]);
        System.out.print("\n Error[1]= "+Error[1]);
    }
}
```

```
else if (images_ke >=100 &&images_ke <= 199)
{
    categories = "beaches";
    if (hasil[1][i] >= 100.0 && hasil[1][i] <= 199.0)
    {
        Score[1]=Score[1]+15-i+1;
        Error[1]=Error[1]+1;
        System.out.print("\n Score[1]= "+Score[1]);
        System.out.print("\n Error[1]= "+Error[1]);
    }
}
```

Proses perhitungan Scoring & Error adalah kelanjutan program looping dari slide sebelumnya. Apabila data gambar testing / Query yang diberikan adalah 60, maka akan masuk ke program "if (images_ke >=0 && images_ke <=99) dan melakukan perhitungan score & error sesuai hasil data peringkat yang didapatkan sebelumnya. Namun, variable error disini =1, karena perhitungan pada looping ini bersifat invers, atau kebalikan. Nantinya (program dibawahnya/keluar dari looping), adalah totalerror= 15-error[1]. Yang artinya, error sebenarnya. Penulisan program ini hanya untuk penyederhanaan program saja.



MENENTUKAN KATEGORI GAMBAR TESTING, SCORING, & ERROR

```
System.out.print("\n Images ke-" + images_ke);  
System.out.print("\n Categories: " + categories);  
TotalError=15-Error[1];  
System.out.print("\n Score: " + Score[1]);  
System.out.print("\n Error: " + TotalError+"\n");
```

Menentukan kategori

Menghitung error

Menampilkan scoring



HASIL PROGRAM

1.0000000000000002	0.8865935255391202	0.8752503213652771	0.8610049454696858	0.8583254301310204	0.8577790402106726	0.8547333740212641	0.854										
313.0	331.0	396.0	840.0	328.0	308.0	320.0	829.0	391.0	385.0	355.0	330.0	305.0	333.0	334.0	381.0	337.0	253.0

Top 15 Images

313.0,

Score[1]= 16

Error[1]= 1

331.0,

Score[1]= 31

Error[1]= 2

396.0,

Score[1]= 45

Error[1]= 3

840.0,

328.0,

Score[1]= 57

Error[1]= 4

308.0,

Score[1]= 68

Error[1]= 5

320.0,

Score[1]= 78

Error[1]= 6

829.0,

391.0,

Score[1]= 86

Error[1]= 7

385.0,

385.0,

Score[1]= 93

Error[1]= 8

355.0,

Score[1]= 99

Error[1]= 9

330.0,

Score[1]= 104

Error[1]= 10

305.0,

Score[1]= 108

Error[1]= 11

333.0,

Score[1]= 111

Error[1]= 12

334.0,

Score[1]= 113

Error[1]= 13

Images ke-313

Categories: buses

Score: 113

Error: 2

Gambar ke-

Data gambar yang tidak memiliki nilai scoring karena berbeda kategori

Nilai Kemiripan/Similarity

Hasil program disamping adalah contoh untuk gambar testing dengan query = 313. Dimana didapatkan kategorinya adalah bus, score = 113 dan Error = 2.



ANALISA

- Praktikum kali ini membahas tentang Image Retrieval berdasarkan isi dari gambar. Image Retrieval pada praktikum ini merupakan suatu metode pencarian citra dengan melakukan perbandingan antara citra query dengan citra yang ada didataset berdasarkan informasi yang ada pada citra tersebut (Query). Dalam metode ini digunakan pencarian berdasarkan kemiripan warna dan fitur.
- Pada hasil perhitungan scoring & error. Dapat diketahui bahwa nilai error beragam, mulai 0 hingga 14, dengan nilai maksimum error yang dapat dihasilkan dalam 1 data testing yaitu 15. Sedangkan nilai scoring juga beragam, dari 16 hingga 130, dari nilai maksimum yang dapat dihasilkan adalah 133. Nilai error yang besar dan nilai scoring yang kecil dari hasil yang telah didapatkan dapat dipengaruhi berbagai faktor seperti warna dan fitur yang kurang lebih hampir mirip, satu dengan yang lainnya. Agar hal tersebut dapat diminimalisir (error kecil & scoring besar), maka bisa dengan menambahkan parameter pencarian.

