

LAMPIRAN A

KODE PROGRAM

Program Utama Pelatihan Citra

% Program Pelatihan Citra

clear;

close all;

clc;

jum_gbr=5;

jum_orang=20;

% Proses Sebelum Pengujian

for indeks_orang=1:jum_orang

indeks_gbr=jum_gbr*(indeks_orang-1)+1:jum_gbr*indeks_orang;

bantu=1;simpan=[];

% Proses baca gambar per orang

for m=indeks_gbr(1):indeks_gbr(jum_gbr)

indeks=num2str(m);

ind=[indeks,'.jpg'];

x=imread(ind);

%figure;imshow(x);title('Gambar masukan (RGB)');

% Proses face detection

[gbr_det]= deteksi_face(x);

% Proses Menyamakan Ukuran Gambar Wajah

[gbr_det]=samakan_ukuran(gbr_det);

% Proses downsample

[gbr_down]=desimasi(gbr_det);

% Proses ubah ke vektor

gbr_vec=reshape(gbr_down,size(gbr_down,1)...

*size(gbr_down,2),1);

% Proses normalisasi vektor (nilai maks = 1)

x1=double(gbr_vec);

[y nmr maks]=ubah_satu(x1);

for n=1:length(y)

if n~=nmr

y(n)=y(n)/maks;

end;

end;

gbr_vec_norm=y;

simpan{bantu}=gbr_vec_norm;

bantu=bantu+1;

end;

gbr_orang{indeks_orang}=simpan;

```

clear simpan;
end;

% Menyesuaikan dengan nama variabel pada paper
for indeks_orang=1:jum_orang
    temp_gbr=gbr_orang{indeks_orang};
    simp=[];
    for pp=1:jum_gbr
        temp_simp=temp_gbr{pp};
        [simp]=[simp temp_simp];
    end;
    X{indeks_orang}=simp;
    clear simp;
end;

% Menyimpan Nilai Xi
save database X jum_orang jum_gbr

```

Program Utama Pengujian Citra

```
close all;
clear all;
clc;

load database.mat
% -----
% Proses Pengujian
masuk = input('Nama File Uji : ','s');
masuk = [masuk, '.jpg'];

% Proses Baca Gambar Uji
gbr_uji = imread(masuk);
figure; imshow(gbr_uji); title('Gambar yang diuji (RGB)');

% Proses Deteksi Wajah Gambar Uji
[gbr_uji_det] = deteksi_face(gbr_uji);

% Proses menyamakan ukuran gambar wajah uji
[gbr_uji_det] = samakan_ukuran(gbr_uji_det);
figure; imshow(gbr_uji_det); title('Gambar yang diuji hasil deteksi wajah');

% Proses downsample gambar uji
[gbr_uji_down] = desimasi(gbr_uji_det);

% Proses ubah ke vektor gambar uji
gbr_uji_vec = reshape(gbr_uji_down, size(gbr_uji_down, 1) ...
    * size(gbr_uji_down, 2), 1);

% Proses normalisasi vektor gambar uji (nilai maks = 1)
x_uji1 = double(gbr_uji_vec);
[y_uji nmr_uji maks_uji] = ubah_satu(x_uji1);

for p = 1:length(y_uji)
    if p ~= nmr_uji
        y_uji(p) = y_uji(p) / maks_uji;
    end;
end;

gbr_uji_vec_norm = y_uji;

% Menyamakan nama variabel sesuai paper
y = gbr_uji_vec_norm;
```

```

% Proses mencari beta estimasi untuk tiap orang (kelas)
% Khusus orang / kelas pertama
X_temp=X{1};
beta_estimasi(:,1)=inv(X_temp'*X_temp)*X_temp'*y;

% Proses mencari y estimasi untuk tiap orang (kelas)
% Khusus orang / kelas pertama
y_estimasi(:,1)=X_temp*beta_estimasi(:,1);

% Proses mencari jarak untuk tiap orang (kelas) dengan y estimasi
% Khusus orang / kelas pertama
jarak(1)=sqrt(sum((y-y_estimasi(:,1)).^2));

% Nilai terdekat sementara
min_jarak=jarak(1);
indeks_min=1;
min_indeks=indeks_min;

for indeks_orang=2:jum_orang
    % Proses mencari beta estimasi untuk tiap orang (kelas)
    % Selain orang / kelas pertama
    X_temp=X{indeks_orang};
    beta_estimasi(:,indeks_orang)=inv(X_temp'*X_temp)*X_temp'*y;

    % Proses mencari y estimasi untuk tiap orang (kelas)
    % Selain orang / kelas pertama
    y_estimasi(:,indeks_orang)=X_temp*beta_estimasi(:,indeks_orang);

    % Proses mencari jarak untuk tiap orang (kelas) dengan y estimasi
    % Selain orang / kelas pertama
    jarak(indeks_orang)=sqrt(sum((y-y_estimasi(:,indeks_orang)).^2));

    if jarak(indeks_orang)<min_jarak
        min_jarak=jarak(indeks_orang);
        min_indeks=indeks_orang;
    end;
end;

indeks_tampil=jum_gbr*(min_indeks-1)+1:jum_gbr*min_indeks;

for mm=1:jum_gbr
    mmm=num2str(indeks_tampil(mm));
    mm_ind=[mmm,'.jpg'];
    xm=imread(mm_ind);
    % figure;imshow(xm);
    % title('Gambar yang mirip dengan gambar uji (RGB) : ada 5 karena 5 gambar
    per orang');

```

```

end;

cek_satu=X{min_indeks};
jarak_satu(1)=sqrt(sum((y-cek_satu(:,1)).^2));

% Nilai terdekat sementara
min_jarak_satu=jarak_satu(1);
indeks_min_satu=1;

for m_satu=2:jum_gbr
    jarak_satu(m_satu)=sqrt(sum((y-cek_satu(:,m_satu)).^2));
    if jarak_satu(m_satu) < min_jarak_satu
        min_jarak_satu=jarak_satu(m_satu);
        indeks_min_satu=m_satu;
    end;
end;

% mmm_satu=num2str((min_indeks-1)*5+m_satu);
mmm_satu=num2str((min_indeks-1)*jum_gbr+indeks_min_satu);
mm_ind_satu=[mmm_satu,'.jpg'];
xm_satu=imread(mm_ind_satu);

% Untuk percobaan 3 (nilai treshold)
if min_jarak_satu > 4.061
    dk= imread('Tidak dikenali.png');
    figure;imshow(dk);
else
    figure;imshow(xm_satu);
    title('Gambar yang mirip dengan gambar uji (RGB) : hanya satu gambar');
end

```

function menyamakan ukuran gambar wajah

```
function[keluar]=samakan_ukuran(masuk)
% Function ini untuk menyamakan ukuran gambar wajah
% Variabel masukan : masuk = gambar wajah yang mau
%                               disamakan ukurannya (grayscale)
% Variabel keluaran : keluar = gambar wajah yang sudah disamakan ukurannya
%                               (grayscale) ke 40 x 40
m=40;
n=40;
keluar=imresize(masuk, [m n]);
```

function deteksi face (butuh file “ facefind.dll ”)

```
function [keluar]= deteksi_face(masuk)
% Function ini untuk melakukan proses deteksi wajah
% Variabel masukan : masuk = wajah yang mau dideteksi
% Variabel keluaran : keluar = cell yang isinya gambar wajah
%                               yang sudah dideteksi
%                               (grayscale)
x=masuk;
if (size(x,3)>1)%if RGB image make gray scale
    try
        x=rgb2gray(x);%image toolbox dependent
    catch
        x=sum(double(x),3)/3;%if no image toolbox do simple sum
    end;
end;

x=double(x);%make sure the input is double format

[output,count,m,svec]=facefind(x);%full scan

N=size(output,2);
col=[1 0 0]; % lihat plotbox.m untuk variabel col
t=2;

if (N>0)
    for i=1:N
        x1=output(1,i);
        x2=output(2,i);
        y1=output(3,i);
        y2=output(4,i);

        vec=[x1 x2 y1 y2];
        ind=find(isinf(vec));%special case if coordinate is Inf
        a=200;%should be realmax, but bug in Matlab? (strange lines occur)
        vec(ind)=sign(vec(ind))*a;

    end;
end;

y=ceil(output);
z=uint8(x);
z1=z(y(3):y(4),y(1):y(2));
% figure;imshow(z1);title('Gambar wajah saja (grayscale)');

keluar = z1;
```

function downsampel citra

```
function[keluar]=desimasi(masuk)
% function ini melakukan proses downsampel
% variabel masukan : masuk = gambar yg mau didownsampel (grayscale)
% variabel keluaran : keluar = gambar yg suda didownsampel (grayscale)

x1 = masuk;

for m=1:size(x1,2)
    if (1+(m-1)*2)<size(x1,2)
        x11(:,m)=x1(:,1+(m-1)*2);
    else
        break;
    end;
end;

for m=1:size(x11,1)
    if (1+(m-1)*2)<size(x11,1)
        x22(m,:)=x11(1+(m-1)*2,:);
    else
        break;
    end;
end;
keluar = x22;
```

function normalisasi vektor

```
function [keluar indeks maks] = ubah_satu(masuk)
% Function ini untuk mengubah nilai max masukan menjadi
% satu dan menemukan indeksinya
% Variabel masukan : masuk = vektor
% Variabel keluaran : keluar = vektor
%                      indeks = indeksinya

keluar=masuk;
maks=masuk(1);
indeks=1;

for k=2:length(masuk)
    if masuk(k)>maks
        maks=masuk(k);
        indeks=k;
    end;
end;

keluar(indeks)=1;
```