

Pemanfaatan Eigenfaces untuk Sistem Pengenalan Wajah (*Face Recognition*)

M. Ibnu Syah Hafizh (13519177)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519177@std.stei.itb.ac.id

Abstract—Pada makalah ini akan dibahas dan diimplementasikan sistem yang mampu mengenali wajah seseorang atau yang dikenal dengan *face recognition* dengan membandingkan struktur wajahnya. Citra yang ini digunakan pada program ini adalah foto individu yang menghadap ke depan untuk membuat representasi dua dimensi dari kepala manusia. Sistem kemudian memproyeksikan citra ke “ruang wajah” (*face space*) yang terdiri dari basis *eigenfaces* yang lengkap. Dikarenakan kesamaan fitur dan bentuk wajah dari tiap individu, citra wajah termasuk dalam wilayah yang relatif kecil dari ruang gambar (*image space*) sehingga dapat direproduksi dengan pengetahuan yang sedikit tentang ruang gambar (*image space*). Ketika citra baru dimasukkan ke dalam sistem ini, sistem ini dapat mengidentifikasi orang dengan tingkat keberhasilan yang relatif tinggi untuk mengidentifikasi dengan benar bahkan pada citra yang mengandung derau.

Keywords—citra; pengenalan wajah; *eigenfaces*

I. PENDAHULUAN

Kemampuan manusia untuk mengenali individu tertentu hanya dengan mengamati wajah manusia cukup luar biasa. Kapasitas ini bertahan bahkan melalui berlalunya waktu, perubahan penampilan dan oklusi parsial. Karena kemampuan luar biasa ini untuk menghasilkan identifikasi positif yang hampir sempurna, banyak perhatian telah diberikan pada metode dimana pengenalan wajah yang efektif dapat direplikasi pada tingkat elektronik. Tentu saja, jika proses rumit seperti identifikasi individu manusia berdasarkan metode non-invasif seperti pengenalan wajah dapat dilakukan secara elektronik, maka bidang seperti keamanan bank dan bandara dapat ditingkatkan secara signifikan, pencurian identitas dapat ditingkatkan lebih lanjut, dikurangi dan keamanan sektor swasta dapat ditingkatkan.

Banyak pendekatan untuk masalah pengenalan wajah secara keseluruhan (*The Recognition Problem*) telah dirancang selama bertahun-tahun, namun salah satu cara paling akurat dan tercepat untuk mengidentifikasi wajah adalah dengan menggunakan apa yang disebut teknik *eigenface*. Teknik *eigenface* menggunakan kombinasi yang kuat dari aljabar linier dan analisis statistik untuk menghasilkan satu set basis *face eigenfaces* terhadap input yang diuji. Proyek ini berupaya mengambil sejumlah besar gambar dari sekelompok orang yang dikenal dan setelah memasukkan gambar wajah yang tidak dikenal, dengan cepat

dan efektif menentukan apakah itu cocok dengan individu yang dikenal atau tidak.

Modul-modul berikut ini akan memberikan panduan bagaimana tepatnya tujuan ini dicapai. Karena ini bukan upaya pertama pengenalan wajah otomatis, penting untuk melihat pendekatan lain apa yang telah dicoba untuk menghargai kecepatan dan keakuratan *eigenfaces*. Ini bukan masalah yang sederhana dan langsung, begitu banyak pertanyaan berbeda yang harus dipertimbangkan saat seseorang mempelajari pendekatan pengenalan wajah ini.

Dengan pemahaman dasar yang tercapai, inilah saatnya untuk hal yang nyata, penerapan prosedur. Ini telah dipecah menjadi langkah-langkah yang lebih kecil dan lebih mudah dikelola. Pertama himpunan basis *eigenface* harus diturunkan dari himpunan citra awal (*Obtaining the Eigenface Basis*). Dengan dasar ini individu yang dikenal dapat diproses untuk membentuk program untuk deteksi dengan menetapkan ambang batas (*Thresholds for Eigenface Recognition*) dan menghitung matriks bobot (*Face Detection Using Eigenfaces*). Akhirnya, dengan program seperti itu, uji kualitas dapat dilakukan untuk menentukan kualitas gambar input yang diperlukan agar identifikasi berhasil dilakukan (*Hasil Tes Deteksi Eigenface*). Dengan cara ini, kesimpulan yang relevan (*Kesimpulan untuk Deteksi Eigenface*) dapat ditarik tentang efektivitas keseluruhan dari metode pengenalan *eigenface*.

II. TEORI DASAR

A. Pengolahan Citra

Pengolahan citra merupakan salah satu cabang dari ilmu komputer yang mempelajari tentang bagaimana program komputer dapat memproses, menganalisis, dan mengolah data visual dari dunia nyata. Citra dapat berupa foto, video, ataupun data sensor yang merepresentasikan objek atau fenomena visual dalam bentuk digital.

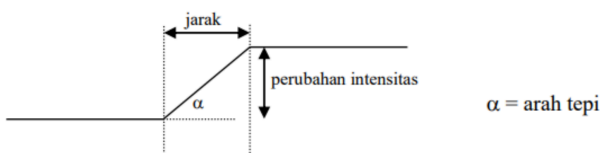
Tujuan dari pengolahan citra adalah untuk mengekstrak informasi yang bermanfaat dari citra tersebut, seperti mengenali pola, mengklasifikasikan objek, atau mengukur ukuran dan posisi objek di dalam citra. Hal ini dapat dilakukan dengan menggunakan algoritma dan metode matematika yang sesuai.

Terdapat berbagai macam aplikasi dari pengolahan citra, seperti sistem pengenalan wajah untuk keamanan, sistem navigasi untuk mobil tanpa supir, hingga sistem monitoring kualitas produk di industri. Teknologi ini juga dapat digunakan dalam bidang kesehatan, seperti dalam diagnosa penyakit dengan menggunakan citra CT scan atau MRI.

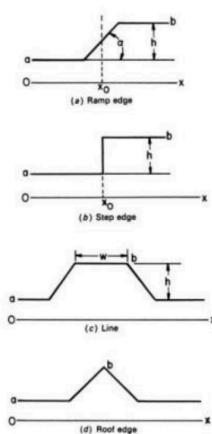
Salah satu teknik yang sering digunakan dalam pengolahan citra adalah deep learning, yaitu sebuah metode yang menggunakan jaringan neural untuk mengenali pola-pola yang terdapat dalam citra. Deep learning memiliki kemampuan yang lebih baik dibandingkan dengan teknik lainnya dalam mengenali pola yang kompleks dan tidak terstruktur, sehingga sering digunakan dalam aplikasi-aplikasi pengolahan citra yang memerlukan tingkat akurasi yang tinggi.

B. Deteksi Tepi

Dalam pengolahan citra, salah satu teknik yang digunakan untuk menentukan batas-batas objek atau tepi di dalam citra tersebut adalah dengan melakukan pendeteksian tepi. Tepi adalah perubahan nilai derajat keabuan yang terjadi secara mendadak dalam jarak yang singkat. Biasanya, tepi berada di batas antara dua daerah pada citra yang memiliki intensitas berbeda yang berubah dengan sangat cepat.



Terdapat empat macam tepi pada citra, yaitu tepi curam, tepi landai, tepi garis, dan tepi atap. Masing-masing jenis tepi ditentukan berdasarkan perubahan intensitas pada bagian tersebut. Berikut dari tiap jenis tepi pada citra.



Pendeteksian tepi dilakukan agar garis batas atau objek dalam citra dapat terlihat lebih jelas. Hasilnya adalah representasi gambar garis dalam citra, biasanya berbentuk citra biner, dengan warna putih menandakan tepi dan hitam menandakan bukan tepi. Hasil pendeteksian tepi kemudian

dapat digunakan untuk berbagai hal seperti mempertajam citra atau mengenali objek dalam citra.

Analisis ataupun pengenalan objek dengan menggunakan pendeteksian tepi terdiri atas tiga tahapan, yaitu: pengekstraksian ciri, segmentasi citra, dan klasifikasi. Tujuan dari analisis ini adalah mengidentifikasi parameter-parameter yang dapat menjadi ciri objek dalam citra. Parameter tersebut kemudian akan digunakan untuk menginterpretasi citra, contohnya adalah dengan melakukan klasifikasi.

Terdapat berbagai teknik yang dapat digunakan dalam pendeteksian tepi. Beberapa teknik yang umum digunakan adalah penapisan citra dengan menggunakan penapis operator gradien dan laplacian. Beberapa operator gradien yang dapat digunakan adalah sebagai berikut.

Operator	Penapis
Gradien	$G_x = \begin{bmatrix} -1 & & \\ & 1 & \\ & & \end{bmatrix}$ $G_y = \begin{bmatrix} & -1 & \\ & & 1 \end{bmatrix}$
Sobel	$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ dan $S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Prewitt	$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ dan $P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Roberts	$R_+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ dan $R_- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
Canny	Menggunakan penapis Gradien, Sobel, Prewitt, atau Roberts.

Operator laplacian merupakan turunan kedua dari gradien. Operator ini dapat digunakan untuk mendeteksi tepi lebih akurat, khususnya pada tepi curam yang berada di persilangan nol. Penggunaannya dapat dilakukan dengan mengkombinasikannya dengan filter gaussian, untuk mengurangi derau pada citra.

Canny adalah operator yang cukup terkenal karena menghasilkan tepi dengan ketebalan 1 pixel. Dalam prosesnya, pertama dilakukan penghalusan citra menggunakan filter gaussian untuk mengurangi derau, dengan nilai sigma tertentu. Setelah itu, dilakukan perhitungan gradien dan arah gradien menggunakan operator gradien lain (Gradien, Sobel, Roberts, atau Prewitt). Lalu setiap pixel dikelompokkan ke dalam pixel tepi atau tidak berdasarkan nilai magnitudo dari gradien. Jika nilainya melebihi nilai ambang, maka pixel tersebut akan menjadi pixel tepi.

Operator Canny dapat mendeteksi dua jenis tepi yaitu tepi kuat dan tepi lemah. Terdapat dua nilai ambang ($T1 < T2$) yang masing-masing menentukan jenis tepi tersebut. Apabila melebihi nilai ambang $T2$, maka pixel termasuk tepi kuat. Pixel lain yang terhubung ke tepi kuat dan memiliki nilai magnitudo melebihi nilai ambang $T1$ akan masuk ke dalam jenis tepi lemah.

C. Pengenalan Wajah (*Face Recognition*)

Pengenalan wajah merupakan teknologi yang memungkinkan sistem untuk mengenali dan memverifikasi identitas seseorang dengan menggunakan fitur-fitur unik dari wajah seseorang. Teknologi ini telah mengalami kemajuan yang signifikan dalam beberapa tahun terakhir, dan telah menemukan banyak aplikasi dalam bidang seperti keamanan, biometrik, dan pemasaran.

Pengenalan wajah terdiri dari dua tahap utama: ekstraksi fitur dan verifikasi identitas. Pada tahap ekstraksi fitur, sistem mengambil citra wajah dan mengidentifikasi fitur-fitur unik seperti bentuk mata, hidung, dan mulut. Fitur-fitur ini kemudian disimpan dalam bentuk vektor yang disebut "deskriptor wajah". Pada tahap verifikasi identitas, program membandingkan deskriptor wajah yang baru diperoleh dengan deskriptor wajah yang tersimpan dalam database untuk memverifikasi identitas seseorang.

Pengenalan wajah menggunakan algoritma yang disebut "pengenalan wajah terstruktur" atau "pengenalan wajah tidak terstruktur". Pengenalan wajah terstruktur menggunakan model wajah yang sudah ditentukan sebelumnya, sedangkan pengenalan wajah tidak terstruktur tidak menggunakan model wajah yang sudah ditentukan sebelumnya dan lebih fleksibel dalam mengenali wajah yang berbeda-beda.

Pengenalan wajah telah menemukan banyak aplikasi dalam bidang keamanan, seperti pengenalan wajah untuk masuk ke gedung atau mengakses perangkat tertentu. Teknologi ini juga telah digunakan dalam bidang biometrik, seperti pemeriksaan paspor atau pengenalan wajah untuk membuka ponsel pintar. Selain itu, pengenalan wajah juga telah digunakan dalam bidang pemasaran, seperti menampilkan iklan yang sesuai dengan demografi sasaran dengan mengenali wajah pelanggan yang datang ke toko.

Namun, pengenalan wajah juga memiliki beberapa masalah yang perlu dipertimbangkan. Misalnya, teknologi ini mungkin tidak selalu akurat dalam mengenali wajah orang yang memiliki fitur wajah yang tidak umum, seperti wajah yang terlalu tua atau muda, atau wajah

D. Eigenface

Eigenface adalah salah satu teknik yang digunakan dalam pengenalan wajah manusia. Teknik ini menggunakan transformasi matematis untuk mengkonversi wajah manusia menjadi representasi matematis yang disebut "eigenwajah". Eigenwajah ini kemudian digunakan untuk mengenali wajah

manusia yang berbeda dengan menggunakan algoritma pembelajaran mesin.

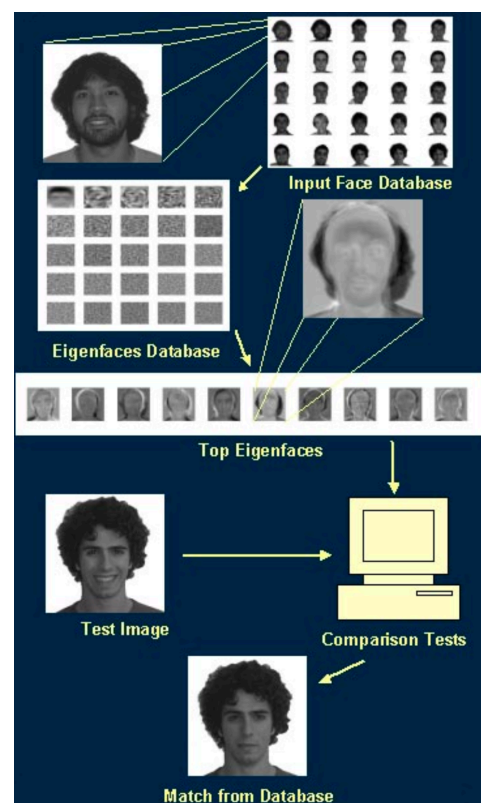
Eigenface dikembangkan pertama kali oleh Matthew Turk dan Alex Pentland pada tahun 1991. Mereka menggunakan algoritma analisis komponen utama (PCA) untuk mengekstrak fitur-fitur yang terdapat pada wajah manusia. PCA merupakan teknik yang digunakan untuk menemukan komponen-komponen utama dari sekumpulan data yang memberikan informasi terbesar tentang data tersebut.

Eigenface bekerja dengan cara mengekstrak fitur-fitur yang terdapat pada wajah manusia seperti garis-garis yang terdapat di sekitar mata, hidung, dan bibir. Fitur-fitur ini kemudian dikonversi menjadi eigenwajah yang merupakan representasi matematis dari wajah tersebut.

Untuk mengenali wajah manusia yang berbeda, eigenface menggunakan algoritma pembelajaran mesin yang disebut k-nearest neighbor (k-NN). Algoritma ini bekerja dengan cara mencari wajah yang paling mirip dengan wajah yang akan diidentifikasi dari sekumpulan data wajah yang telah dikenali sebelumnya.

III. RANCANGAN SISTEM YANG DIBANGUN

Sistem pengenalan wajah eigenface dapat dibagi menjadi dua bagian utama: pembuatan basis eigenface dan pengenalan wajah baru. Sistem ini memiliki *flow* sebagai berikut:



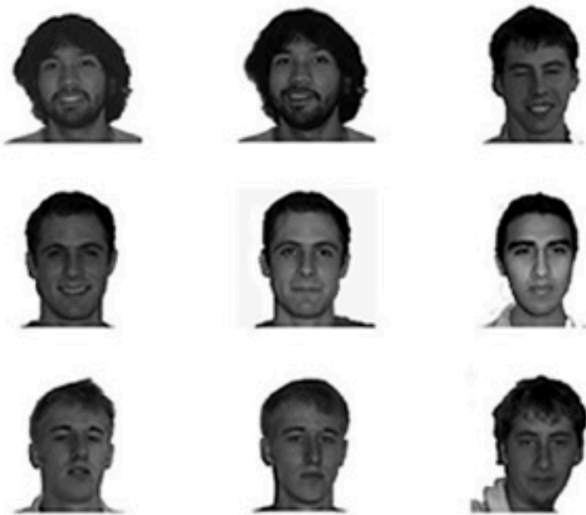
Gambar III.1. Alur pengenalan wajah menggunakan Eigenface

A. Pembuatan Basis Eigenface

Teknik eigenface adalah salah satu alternatif solusi yang cukup efektif dan sederhana untuk permasalahan pengenalan wajah. Faktanya, *eigenface* adalah cara paling intuitif untuk mengklasifikasikan wajah. Teknik eigenface menggunakan lebih banyak informasi dengan mengklasifikasikan wajah berdasarkan pola wajah secara umum. Pola-pola ini mencakup, namun tidak terbatas pada ciri-ciri khusus dari wajah. Dengan menggunakan lebih banyak informasi, analisis eigenface secara alami lebih efektif daripada pengenalan wajah berbasis fitur seperti yang digunakan pada teknik yang konvensional.

Eigenfaces pada dasarnya tidak lebih dari vektor basis untuk wajah asli. Ini dapat dikaitkan langsung dengan salah satu konsep paling mendasar yakni Analisis Fourier. Analisis Fourier mengungkapkan bahwa jumlah sinusoid berbobot pada frekuensi yang berbeda dapat menyusun ulang sinyal dengan sempurna! Dengan cara yang sama, sejumlah eigenface berbobot dapat merekonstruksi wajah orang tertentu dengan mulus. Menentukan basis eigenfaces ini adalah inti dari teknik eigenface.

Sebelum menemukan basis eigenfaces, pertama-tama harus mengumpulkan satu set citra wajah. Citra wajah ini menjadi basis data wajah yang dikenal (*supervised*). Kemudian, sistem akan menentukan apakah wajah yang tidak dikenal cocok atau tidak dengan salah satu wajah yang dikenal ini. Semua gambar wajah harus memiliki ukuran yang sama (dalam piksel), dan untuk kasus ini, harus dalam citra *grayscale*, dengan nilai mulai dari 0 hingga 255. Setiap gambar wajah diubah menjadi vektor Γ_n dengan panjang N ($N = \text{imagewidth} * \text{imageheight}$). Kumpulan wajah yang paling berguna memiliki banyak gambar per orang. Ini secara tajam meningkatkan akurasi, karena meningkatnya informasi yang tersedia pada setiap individu yang dikenal. Kami akan menyebut koleksi wajah kami sebagai ruang wajah. Ruang ini berdimensi N.



Gambar III.1. Contoh isi data set citra wajah

Selanjutnya perlu menghitung rata-rata wajah dalam ruang wajah (*face space*). Di sini M adalah jumlah wajah di set yang digunakan:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$



Gambar III.1. Contoh rata-rata citra wajah

Kemudian menghitung perbedaan setiap wajah dari rata-rata citra wajah yang telah didapatkan dengan rumus :

$$\Phi_i = \Gamma_i - \Psi$$

Dengan menggunakan perbedaan ini, dihitung matriks kovarians (C) untuk dataset yang digunakan. Kovariansi antara dua set data mengungkapkan seberapa besar korelasi set tersebut.

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = \frac{1}{M} \sum_{n=1}^M \begin{pmatrix} \text{var}(p_1) & \dots & \text{cov}(p_1, p_N) \\ \vdots & \ddots & \vdots \\ \text{cov}(p_N, p_1) & \dots & \text{var}(p_N) \end{pmatrix} = AA^T$$

$A = [\Phi_1 \Phi_2 \dots \Phi_M]$ dan p_i = piksel i dari panjang n .

Wajah eigen yang dicari sebagai basis eigenface adalah vektor eigen dari C. Secara teoritis, karena C berdimensi N (jumlah piksel dalam citra), pemecahan wajah eigen memburuk dengan sangat cepat dan pengenalan wajah Eigenface tidak akan mungkin dilakukan. Namun, faktanya teknik eigenface memiliki karakteristik yang dapat mengatasi permasalahan ini yakni dengan menyederhanakan basis eigendace awal.

Berdasarkan teknik Principal Component Analysis (PCA), jumlah vektor dapat dikurangi untuk matriks kovarians dari N (jumlah piksel dalam citra) menjadi M (jumlah gambar dalam kumpulan dataset citra). Secara umum, PCA digunakan untuk mendeskripsikan ruang berdimensi besar dengan sekumpulan vektor yang relatif kecil. Ini adalah teknik populer untuk menemukan pola dalam data berdimensi tinggi, dan digunakan secara umum baik dalam pengenalan wajah maupun kompresi gambar. PCA dapat diterapkan untuk pengenalan wajah karena gambar wajah biasanya memiliki tingkat kemiripan yang tinggi satu sama lain (relatif terhadap gambar non -wajah) dan dengan jelas berbagi pola dan struktur umum yang sama.

B. Pengenalan Wajah menggunakan Eigenface

Eigenface menjadi metode yang cukup mudah, ekonomis secara komputasi, dan cukup efektif untuk menentukan apakah wajah yang diberikan adalah orang yang dikenal (yang terdapat di dalam data set citra), citra wajah baru, atau bukan wajah sama sekali. Himpunan vektor eigenface dapat dianggap sebagai himpunan basis bebas linier untuk ruang muka. Setiap vektor hidup dalam dimensinya sendiri, dan satu set permukaan eigen M akan menghasilkan ruang berdimensi M.

Perlu juga diketahui bahwa eigenface mewakili komponen utama dari set citra wajah. Komponen utama ini sangat berguna dalam mempermudah proses pengenalan suatu kumpulan data. Untuk membuatnya lebih sederhana, pertama-tama ambil semua citra rata-rata yang dikurangi dalam database dan proyeksikan ke ruang wajah (*face space*). Pada dasarnya adalah perkalian titik dari setiap citra wajah dengan salah satu basis eigenface. Dengan menggabungkan vektor sebagai matriks, maka didapatkan matriks bobot ($M \times N$, dengan N adalah jumlah citra dalam database).

$$\omega_k = \mu_k (\Gamma_{new} - \Psi)$$

$$\Omega^T = [\omega_1 \omega_2 \dots \omega_M]$$

$$WeightMatrix = \begin{pmatrix} \omega_{11} & \dots & \omega_{1n} \\ \vdots & \ddots & \vdots \\ \omega_{m1} & \dots & \omega_{m'n} \end{pmatrix}$$

Citra input juga dapat diproyeksikan ke ruang wajah (*face space*). Hasil proyeksi ini akan menghasilkan vektor dalam ruang dimensi M. M adalah jumlah eigenfaces yang digunakan. Secara logika, wajah orang yang sama akan dipetakan cukup dekat satu sama lain di ruang wajah ini. Pengenalan wajah dapat dilakukan dengan menemukan citra yang terdapat pada database, atau secara matematis menemukan jarak Euclidean minimum antara titik uji dan titik basis data.

$$\epsilon_k = \sqrt{|\Omega_{new} - \Omega_k|^2}$$

Karena kesamaan keseluruhan dalam struktur wajah, piksel wajah mengikuti distribusi wajah secara keseluruhan. Kombinasi distribusi ini dan analisis komponen utama memungkinkan reduksi dimensi, di mana hanya beberapa eigenface pertama yang mewakili informasi dalam sistem. Selain itu, kompleksitas komputasi menjadi sangat berkurang. Dalam sistem yang akan dibangun, digunakan dua teknik untuk pengenalan gambar.

Teknik pertama adalah *Average Technique* atau teknik rata-rata. Dalam database, semua vektor bobot citra dari orang yang sama dirata-ratakan bersama. Hal ini akan menciptakan "kelas wajah" di mana matriks bobot yang lebih kecil mewakili wajah umum dari keseluruhan sistem. Saat mendapat input citra wajah baru, vektor bobotnya dibuat dengan memproyeksikannya ke ruang wajah. Wajah tersebut kemudian dicocokkan dengan

kelas wajah yang meminimalkan jarak euclidean. 'Hit' dihitung jika citra yang diinput cocok dengan kelas wajahnya. Error dapat terjadi jika jarak minimum cocok dengan kelas wajah orang lain. Misalnya, sebuah database memiliki total empat ratus gambar, terdiri dari empat puluh orang dengan masing-masing sepuluh gambar. Teknik rata-rata menghasilkan matriks bobot dengan empat puluh vektor (empat puluh kelas wajah yang berbeda).

Teknik kedua yang digunakan adalah *Removal Technique*. Prosedur ini sedikit berbeda dari teknik rata-rata dalam satu cara utama. Matriks bobot mewakili vektor proyeksi gambar untuk gambar dari seluruh database. Untuk hasil empiris, citra dihapus dari program, dan kemudian diproyeksikan ke ruang wajah. Vektor bobot yang dihasilkan kemudian dibandingkan dengan vektor bobot semua citra. Citra tersebut kemudian dicocokkan dengan citra wajah yang meminimalkan jarak euclidean. Perbedaan utama antara teknik penghapusan dan teknik rata-rata adalah jumlah kemungkinan gambar yang dapat dicocokkan dengan wajah uji yang masih akan menghasilkan error. Untuk sebuah database, matriks bobot dengan empat ratus vektor, namun citra baru yang diterima dapat berpotensi error hingga sepuluh wajah berbeda.

IV. IMPLEMENTASI DAN HASIL

Pada implementasi sistem pengenalan wajah menggunakan eigenface kali ini, terdapat tiga file utama yakni loadSubset.m, face_recognition.m, dan main_facerec.m. Data set citra yang digunakan untuk model pembelajaran terdapat pada folder yBaleFace.

Pada file loadSubset.m akan dilakukan load sebuah atau beberapa set subset dari database dengan cara membuat variable yang bernama imgs dan labels kosong untuk diisi pada setiap iterasi pembacaan file. Variabel tersebut akan dimasukkan dengan citra dari yang sudah dimasukkan dalam bentuk matriks.

Berikut adalah potongan source code pada file loadSubset.m:

```
function [ imgs, labels ] = loadSubset(subsets, yaleDir)
if nargin < 2
    yaleDir = '/Users/ibnuhafizh/Desktop/yaleBfaces/yaleBfaces';
end
imgs = [];
labels = [];
for i=1:length(subsets)
    subsetDir = sprintf('%s/subset%d/', yaleDir, subsets(i));
    files = dir(subsetDir);
    for j=1:length(files)
        [person, count] = sscanf(files(j).name, 'person%02d');
        if count == 1
            img = single(im2double( ...
                adaphisteq(imread([subsetDir files(j).name]))) ...
            ));
            imgs = [imgs; img(:)'];
            labels = [labels; person];
        end
    end
end
imgs = imgs';
end
```

Pada file loadSubset.m akan dilakukan load sebuah atau beberapa set subset dari database dengan cara membuat variable yang bernama imgs dan labels kosong untuk diisi pada setiap iterasi pembacaan file. Variabel tersebut akan

dimasukkan dengan citra dari yang sudah dimasukkan dalam bentuk matriks.

Pada file ini akan mengeluarkan output berupa variable imgs yaitu matrix of images berukuran dxN dimana N adalah jumlah gambar dan d adalah jumlah pixel di setiap gambar. Selain itu program juga akan mengeluarkan variable labels yaitu sebuah vector dengan panjang N yang akan menyimpan ID dari setiap orang.

Berikut adalah source code face_recognition.m:

```
function [acc,mindist,recog] = face_recognition(w, labels, w2, labels2,N)
%% Initializations
v=w; % v contains the training set.
% N = Number of eigenface components used for each image
%% Subtracting the mean from v
0=single(ones(1,size(v,2)));
m=single(mean(v,2)); % m is the mean of all images.
vzm=v-(m*0); % vzm is v with the mean removed.

%% Calculating eigenvectors of the correlation matrix
L=single(vzm)*single(vzm)';
[V,D]=eig(L);
V=single(vzm)*V;
% Pick the eigenvectors corresponding to the N largest eigenvalues.
V=V(:,end:-1:end-(N-1));
%% Calculating the signature weight for each image
cv=zeros(size(v,2),N);
for i=1:size(v,2)
% Each row in cv is the signature for one image.
cv(i,:)=single(vzm(:,i))*V;
end
%% Recognition
% run algorithm and see if can correctly recognize the face.
recog = [];
dist = [];
mindist = [];

for j=1:size(w2,2)
r=w2(:,j); % r contains a test image
p=r-m; % Subtract the mean
s=single(p)*V;
z=[];
for i=1:size(v,2)
z=[z,norm(cv(i,:)-s,2)];
end
dist = [dist;z];
[a,i]=min(z);
mindist = [mindist,a];
recog = [recog,labels(i)];
end

for j=1:length(recog)
% result=1 if recog match labels2, 0 otherwise
result(j) = isequal(recog(j), labels2(j));
end

acc = sum(result)./length(result); %accuracy
end
```

Pada source code diinisialisasi variabel-variabel antara lain:

- w : training set
- labels : training set labels
- w2 : test set,
- labels2 : test set labels
- N : number of eigenface components used (starting from the first eigenface corresponding to the highest eigenvalue)
- acc : accuracy (number of correct recognition/ total test set entries)
- mindist : minimum distance of a test set entry to a training set entry

- recog : recognition label, i.e the label of a training set entry which provides minimum distance to one test set entry

Pada awalnya akan dilakukan inisialisasi dengan mengisi variable v dengan w yaitu training set. Setelah itu, training set yang telah disimpan pada v akan dikurangi dengan jumlah mean dari eigenfaces yang akan disimpan pada variable vzm.

Setelah itu akan dilakukan kalkulasi eigenvectors dari korelasi matriks dengan mengambil N eigenfaces berdasarkan dari N terbesar dari eigenvalues. Lalu dilakukan kalkulasi signature weight dari setiap citra.

Lalu dilakukan proses recognition dengan membuat variable kosong bernama recog, dist, mindist. Pada bagian recognition akan dilakukan iterasi untuk mengurangi test image dari mean. Setelah itu, akan dicari min distance dan dihitung akurasinya.

Adapun program main_facerec.m adalah sebagai berikut :

```
%% Face recognition
% This algorithm uses the eigenface system (based on principal component
% analysis - PCA) to recognize faces. For more information on this method
%% Clear everything before starting
clear all;close all;clc;

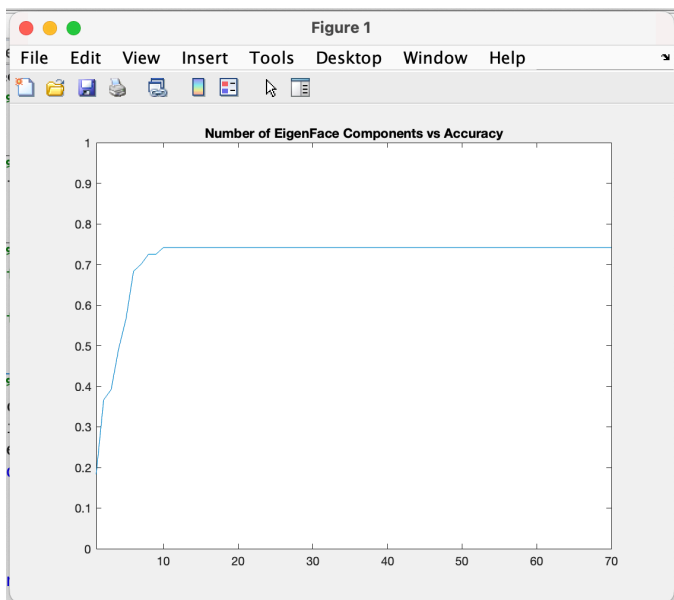
%% Loading the datasets into matrix w
%training set of 10 subjects x 7 lighting conditions x (50x50 pixels) image
[ w, labels ] = loadSubset(0);
%training set of 10 subjects x 7 lighting conditions x (50x50 pixels) image
[ w2, labels2 ] = loadSubset(1);

%% Perform face recognition with varying N (number of eigenface components)
Acc=[];
Mindist=[];
Recog=[];
for N=1:size(w,2)
[acc, mindist, recog] = face_recognition(w, labels, w2, labels2,N);
Acc= [Acc, acc];
Mindist= [Mindist; mindist];
Recog = [Recog; recog];
end

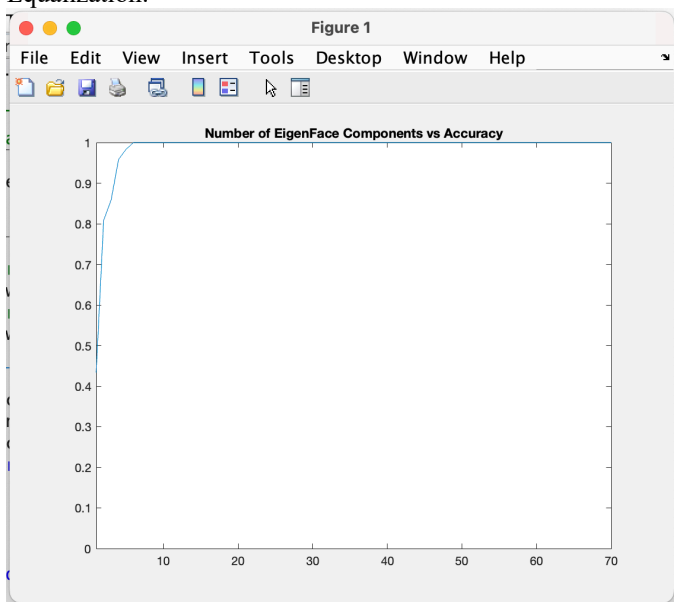
figure, plot(1:size(w,2),Acc);
axis([1 size(w,2) 0 1]);
title ('Number of EigenFace Components vs Accuracy');
```

Source code diatas adalah main program dari project ini. Pada source code ini akan dilakukan loading dataset dan training untuk 10 subject dari kondisi pencahayaan yang berbeda. Setelah itu, akan dieksekusi program face recognition dengan N yang bervariasi. Jika dijalankan maka program akan mengeluarkan output sebagai berikut,

Tanpa preprocessing dengan Teknik Adaptive Histogram Equalization:



Dengan *preprocessing* dengan Teknik Adaptive Histogram Equalization:



Pada output didapat akurasi berbeda-beda sesuai dengan jumlah komponen eigenface, makin besar jumlah komponen eigenface maka akan semakin besar akurasi dari face recognitionnya. Pada akhir grafik terlihat Untuk hasil tanpa *preprocessing* ketika jumlah komponen eigenface mencapai 15 maka akurasi yang muncul stagnan di nilai 75%, sedangkan untuk hasil dengan *preprocessing* ketika jumlah komponen eigen 15 maka akurasi yang muncul sebesar sekitar 100%.

Besar akurasi 100% ini merupakan peningkatan jika dibandingkan dengan kode sebelumnya jika belum dilakukan pre-processing. Pada kode sebelumnya, peak accuracy sebesar 75% karena gambar yang diperoleh masih ada illumination problem. Hal yang dilakukan untuk menaikkan jumlah akurasi adalah dengan melakukan Teknik Adaptive Histogram

Equalization yaitu teknik pemrosesan gambar digital yang digunakan untuk meningkatkan kontras gambar. Ini berbeda dari pemerataan histogram normal dalam hal metode adaptif meningkatkan kontras secara lokal. Ini membagi gambar menjadi blok yang berbeda dan menghitung pemerataan histogram untuk setiap bagian. Jadi, AHE menghitung banyak histogram, masing-masing sesuai dengan bagian gambar yang berbeda. Ini meningkatkan kontras lokal dan definisi tepi di semua wilayah gambar yang berbeda.

V. KESIMPULAN

Berdasarkan percobaan yang telah dilakukan maka dapat disimpulkan bahwa:

- Analisis teknik pengenalan eigenface menggunakan metode average dan removal memberikan bukti bahwa metode tersebut membuktikan akurat bergantung dengan dataset dan kualitas gambarnya.
- Pada Teknik eigenface ini untuk dataset yang ada pada kelas mempunyai akurasi sekitar 75% namun setelah dilakukan preprocessing dengan Teknik Adaptive Histogram Equalization meningkat menjadi 100%.
- Kualitas dataset kelas ini sangat mempengaruhi ketika diinput ke dalam algoritma eigenface. Kualitas dataset yang bagus akan menghasilkan akurasi yang tinggi.

TAUTAN LINK PROGRAM

https://github.com/ibnuhafizh/FaceRecognition_using_Eigenfaces

UCAPAN TERIMA KASIH

Segala puji bagi Allah SWT yang telah memberi kemudahan dan kelancaran dalam menyusun makalah ini sehingga dapat diselesaikan dengan baik dan tepat waktu. Terima kasih kepada kedua orang tua dan teman-teman yang memberi dukungan secara moral maupun doa selama penyusunan makalah. Terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T selaku dosen mata kuliah Interpretasi dan Pengolahan Citra yang telah memberi ilmu yang menjadi pondasi dasar dari makalah ini. Disadari masih terdapat kekurangan dan kesalahan kata dalam makalah ini, diharapkan makalah ini dapat berguna dan bermanfaat serta dikembangkan lebih jauh sehingga berdampak untuk masyarakat luas.

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/01-Pengantar-Pengolahan-Citra-Bag1-2021.pdf>. Diakses pada 18 Desember 2022.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/18-Pendeteksian-Tepi-Bagian1-2022.pdf>. Diakses pada 18 Desember 2022
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/19-Pendeteksian-Tepi-Bagian2-2022.pdf>. Diakses pada 18 Desember 2022
- [4] <https://towardsdatascience.com/eigenfaces-recovering-humans-from-ghosts-17606c328184>. Diakses pada 18 Desember 2022

- [5] Matthew A. T. & Alex P. P. (1991). *Eigenface for recognition*. Journal of Cognitive Neuroscience.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 2022.



M. Ibnu Syah Hafizh