Pengenalan wajah dengan algorithma Eigen Face

Oleh: Hanif Al Fatta

Abstraksi

Pengenalan wajah (face recognition) yang merupakan salah satu penerapan image processing, kini telah dipakai untuk banyak aplikasi. Pada tulisan ini akan dibahas salah satu teknik pengenalan wajah dengan algoritma eigen face. Eigen face digunakan untuk mereduksi vektor image menjadi vektor yang lebih sederhana yang dinamakan eigen vector. Pada tulisan ini aplikasi akan dibatasi pada kondisi lingkungan yang terkontrol (controlled environment) dimana input image yang diproses merupakan image dengan ukuran 78 x 78 pixel dengan tipe grayscale.

Kata kunci*: face recognition, eigen face*

1. Pendahuluan

Face recognition sekarang telah dikembangkan untuk banyak aplikasi terutama untuk aplikasi-aplikasi keamanan. Penggunaan wajah sebagai identifier mempunyai banyak manfaat, terutama kepraktisannya karena tidak perlu dibuat kartu atau photo untuk identifikasi. Masalah utamanya adalah sebuah image yang mewakili sebuah gambar terdiri dari vektor dengan ukuran relatif besar. Ada banyak teknik untuk mereduksi dimensi dari image yang akan diproses, salah satunya yang akan dibahas disini dengan menggunakan eigen face algorithm.

2. Eigen Face

Eigen face adalah salah satu algoritma pengenalan wajah yang berdasarkan pada Principle Component Analysis (PCA) yang dikembangkan di MIT. Algoritma Eigen Face secara keseluruhan cukup sederhana. Training Image direpresentasikan dalam sebuah vektor flat (gabungan vektor) dan digabung bersama-sama menjadi sebuah matriks tunggal. Eigen Vector kemudian diekstraksi dan disimpan dalam file temporary atau database. Training image kemudian kemudian diproyeksikan dalam feature space, dinamai face space yang ditentukan oleh eigen vektor. Adapun algoritma selengkapnya adalah:

1. Buat MakeEigenVectors(ImageList, N, M): Image List adalah kumpulan dari N training image, dimana setiap image adalah W x H pixel. M adalah jumlah eigen vektor yang harus dibuat.
2. Gabungkan setiap image dalam WH elemen vektor dengan menggabungkan semua baris. Buat ImageMatrix sebagai matrik N x WH berisi semua gambar yang digabung

3. Jumlahkan semua baris pada ImageMatrix dan bagi dengan N untuk mendapatkan rataan gambar gabungan. Kita namakan vektor elemen WH ini dengan ψ.
4. Kurangi ImageMatrix dengan average image ψ. Kita namakan matriks baru ukuran N x WH sebagai Φ.
5. Hitung dot product dari semua kemungkinan pasangan gambar. Kita mendapatkan L sebagai matrix ukuran N x N dimana L[i][i]=dot product dari Φ[i] dan Φ[j].
6. Hitung N eigen value dan eigen vektor yang bersesuiain dari L. ambil eigen vektor M yang memiliki eigen value tertinggi. Setiap Eigen Vektor akan memilki panjang N elemen
7. Lakukan perkalian matrik dari setiap M eigen vektor yang terpilih dengan Φ dan simpan hasilnya yaitu matrik yang berukuran 1 x WH. Simpan juga hasil rataan ψ.

Kemudian lakukan proyeksi dengan algoritma sebagai berikut:

1. projectToFaceSpace(image): image berukuran W x H pixel
2. kita gabung elemen vektor WH dan kita sebut img
3. load nilai rataan ψ dan EigenMatrix dari database atau file
4. kurangi img dengan ψ, kita dapatkan img'
5. lakukan dot product antara img' dengan ψ untuk mendapatkan vektor ukuran M img''
6. hitung nilai norm =

$$norm = \sqrt{\sum_{i=1}^{iL} img''[i] \times img''[i]}$$

bagi setiap elemen pada img'' dengan norm, di dapat face space dari gambar.

Learning adalah proses untuk memproyeksikan semua wajah yang dikenal ke face space dan meyimpan hasil representasinya serta mengidentifikasi setiap orang .

LearnFace(ImageList,N,M): Image List adalah kumpulan dari N training image, dimana setiap image adalah W x H pixel. M adalah jumlah eigen vektor yang harus dibuat.

1. Panggil MakeEigenVector(ImageList,N,M)
2. Untuk setipa gambar pada imageList, panggil projectTofaceSpace(image) dan simpan hasilnya pada database.

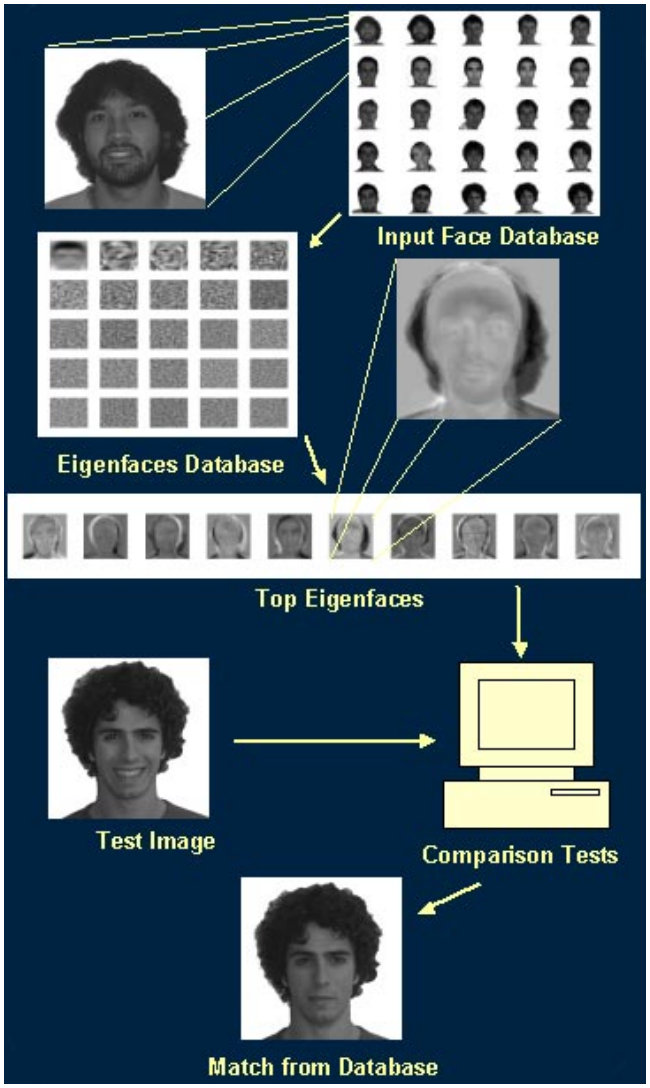
Proses terakhir adalah identifikasi, yaitu memproyeksikan test image ke face space dan menghitung score.

1. Load semua wajah yang sudah diproyeksikan dari database
2. Proj=projectToFaceSpace(image)
3. Lakukan dot product antara proj dengan semua wajah yang telah diproyeksikan. Hasilnya adalah score.
4. Ambil Nilai yang tertinggi sebagai hasil tertinggi dari wajah yang telah diproyeksikan , wajah ini senagai hasil identifikasi.

3. Rancangan Sistem

3.a Basis Input face

Pengenalan wajah dengan eigen face dapat dibagi menjadi 2 tahapan utama pertama, membuat basis eigen face dan pengenalan wajahnya. Untuk lebih jelasnya dapat dilihat pada flowchart berikut ini :



Gambar 1: Flow Chart dari proses pengenalan wajah dengan algoritma Eigen face

Input Face database yang digunakan adalah image dengan ukuran yang sama yaitu 78 x 78 pixel dengan kualitas warna grayscale. Input face database yang digunakan terdiri dari beberapa orang, dimana masing-masing orang mempunyai 2-4 pose yang berbeda-beda. Untuk lebih jelasnya perhatikan contoh berikut:

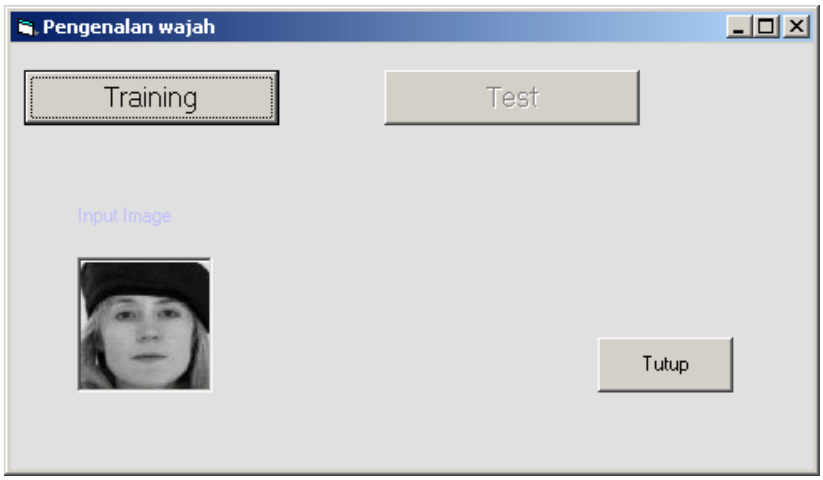
Gambar 2: 4 pose dari 1 orang yang sama

Penggunaan beberapa pose untuk 1 objek dimaksudkan agar proses identifikasi bisa mengenali objek tersebut meskipun objek menggunakan beberapa ekspresi misalnya senang atau sedih.

3.b Proyeksi basis input face ke basis eigen face

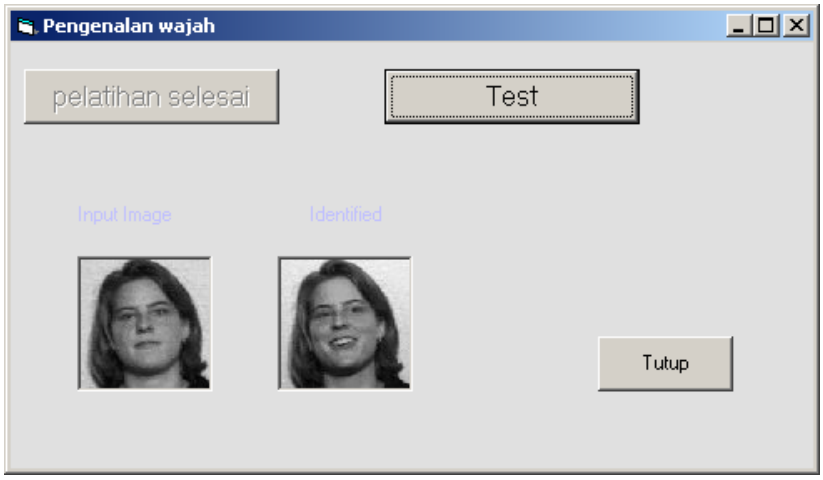
Untuk mendapatkan basis database eigen face dari input face yang ada digunakan algoritma yang telah dijelaskan pada bagian 2. Aplikasinya dituangkan dalam sebuah class yang dibuat dengan bahasa Visual Basic. Selengkapnya dapat dilihat pada lampiran dengan nama classImageProcessing. Sedangkan untuk proses matching input dengan databasenya (identifikasi) ditangani oleh ClassFaceRecognizer. Adapun input dilakukan secara otomatis (program men-scan database wajah dan mengambil input face secara sekuensial) untuk dicocokkan dengan semua eigen face yang telah dihitung.

4. Implementasi



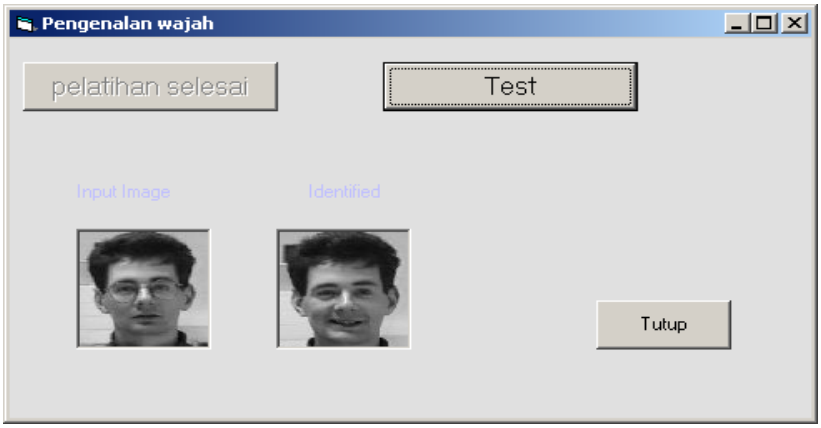
gambar 3: tampilan aplikasi pengenalan wajah

Button training digunakan untuk memanggil classImageProcessing yang akan mendapatkan basis eigen face dari semua input face yang ada dalam database (dalam implementasi ini disimpan dalam file). Ketika semua face telah diproyeksikan Button test akan aktif dan siap untuk proses identifikasi wajahnya.



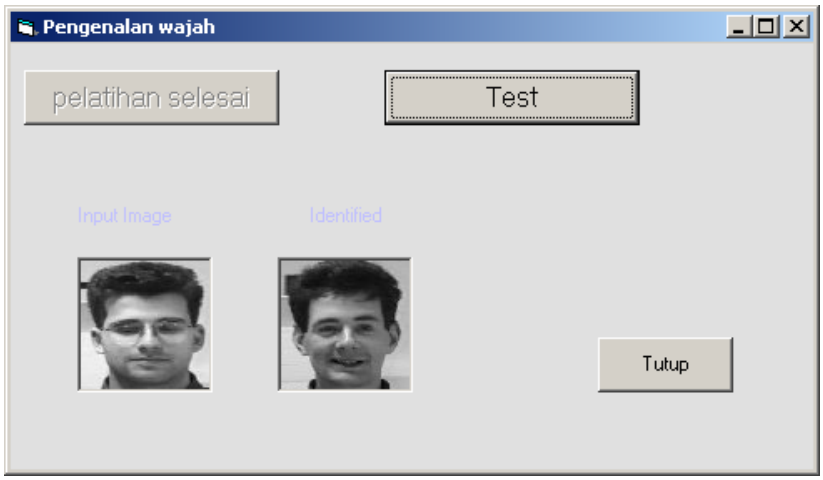
Gambar 4: Hasil identifikasi wajah yang benar

hasil dari test menunjukkan dengan input image yang mempunyai ekspresi biasa, sistem bisa mencocokkan objek wajah yang sama dengan ekspresi yang berbeda (tertawa.). Sistem bisa juga membedakan perubahan wajah seseorang karena pemakaian aksesoris seperti kacamata.



gambar 5: objek dengan kacamata masih bisa dikenali

Tetapi Sistem kadang salah mengidentifikasi kalau ada 2 orang yang mempunyai wajah hampir sama.



Gambar 5: hasil identifikasi yang salah

5. Kesimpulan

Kesimpulan yang dapat diambil diantaranya adalah :

1. Pengenalan wajah dengan algoritma Eigenface bisa digunakan untuk mengidentifikasi wajah meskipun objek yang diidentifikasi menampilkan ekspresi wajah yang berbeda-beda.
2. Pada penelitian ini belum diukur berapa persen tingkat kesalahan yang dilakukan oleh sistem

Daftar pustaka

Anonim, *Eigen Faces*, http://www.stanford.edu/~mbinu/perception/node48.html

Anonim, *eigenface and codification*, http://sirio.psi.ucm.es/PROYECTOS/EIGENCAR/eiweb1_e.html

Dimitri Pissarenko, *Eigen face Based face Recognition,* http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html

Jon Krueger, Marshal Robinson, Doug Kochelek, Matthew Escara: *Obtaining the Eigenface Basis*, http://cnx.rice.edu/content/m12531/latest/

ClassImageProcessing and ClassFaceRecognition downloaded from: http://www.fuzzgun.btinternet.co.uk/