
PEMANFAATAN RASPBERRY PI SEBAGAI PROCESSOR PADA PENDETEKSIAN DAN PENGENALAN POLA WAJAH

Fadli Sirait, Yoserizal

Program Studi Teknik Elektro, Fakultas Teknik
Universitas Mercu Buana, Jakarta, Indonesia
fadlisirait@gmail.com, yoser174@gmail.com

Abstrak - Teknologi biometrik adalah teknologi untuk mengidentifikasi makhluk hidup. Tujuan perancangan adalah untuk membangun sistem pendeteksi wajah dari objek citra yang didapat dari gambar frame video melalui kamera. Kemudian dilakukan pendeteksi pola wajah yang dikenali dan mencari kemiripan terhadap database model wajah menggunakan Raspberri Pi berbasis penggunaan perangkat lunak *Free* dan *Opensource*. Perancangan ini menggunakan metode pengenalan objek citra wajah dengan Haar Cascade Classifier yang diimplementasikan pada library OpenCV, sedangkan metode pengenalan pola wajah dengan menggunakan analisa PCA (*Principal Component Analysis*) dan LDA (*Linear Discriminant Analysis*) menggunakan pemrograman perangkat lunak yang dibuat berbasis Python. Perangkat lunak yang dikembangkan juga dijalankan pada sistem operasi berbasis Linux Raspbian (Jessie dan Wheezy) yang diinstal di Raspberry Pi. Proses input citra menggunakan USB kamera yang dipasang pada Raspberry Pi 2 Model B yang dilengkapi dengan LCD 3,5 inchi. Berdasarkan data pengujian terhadap 127 input didapat tingkat akurasi untuk pendeteksi satu objek wajah 84-97% sementara performa penggunaan CPU pada Raspberry Pi 41.87-46.25%.

Keywords: *face detection and recognition, Raspberry pi, pengolahan citra, embedded system.*

PENDAHULUAN

Ada banyak (banyak digunakan di dunia bisnis, di perusahaan apa saja, bisa disebutkan, bahasa *kuantitatif/* terukur) teknologi yang identifikasi yang sudah ada di dunia bisnis dan sudah digunakan secara komersil sejak bertahun-tahun yang lalu, metode yang paling sering digunakan misalnya dengan system password/PIN (Personal Identification Number) dan dengan system token. Sistem ini masih memiliki celah untuk disalahgunakan seperti dicuri, dipalsukan.

Dewasa ini (*dibuatkan tahun proses pengembangan, dan implemetasinya, sumber data informasinya*) telah banyak dikembangkan system dengan identifikasi biometrik yang menggunakan teknik pengenalan pola untuk identifikasi seseorang menggunakan karakteristik fisiologi orang tersebut seperti pengenalan sidik jari, dan pengenalan retina mata.

Teknologi biometrics penggunaannya mengharuskan seseorang memposisikan tubuh mereka pada posisi tertentu yang sesuai dengan posisi sensor yang membuat teknologi ini terkesan kaku.

Teknologi biometrics mengharuskan orang tersebut untuk tidak bergerak dalam beberapa waktu tertentu selama proses identifikasi untuk membuat sistem pembacaannya akurat. Penggunaan teknologi ini sangat cocok untuk diimplementasikan pada sistem identifikasi yang membutuhkan sekuriti yang tinggi.

Penggunaan identifikasi yang lebih fleksibel misalnya pada pengenalan pelanggan sebuah toko atau dalam rumah untuk mengidentifikasi siapa saja yang tinggal didalamnya dibutuhkan teknologi pengenalan dengan wajah dan atau suara untuk menuju smart-environment lebih mudah dan tepat digunakan. Tanpa membutuhkan sensor elektro-magnetik, tidak mengharuskan seseorang untuk memposisikan tubuhnya pada posisi tertentu

Penerapan identifikasi wajah dibutuhkan komputer untuk mengolah sinyal digital dari foto wajah seseorang kemudian mencocokkannya dengan yang ada di database. Proses peidentifikasi ini membutuhkan komputer yang relatif cepat

untuk memendekan waktu bebannya. Penggunaan personal komputer memiliki kekurangan dalam hal power dan dimensi yang relative lebih besar dan memakan tempat yang banyak, alternatif penghematan energi. Dewasa ini banyak digunakan sistem yang tertanam/embedded yang dapat menggantikan komputer personal seperti salah satunya adalah *Raspberry pi*.

KONSEP PEMOGRAMAN

Percobaan pertama dengan tujuan untuk membaca database gambar dari lokasi direktori yang disiapkan yang kemudian diproses dengan menggunakan PCA untuk mendapatkan *eigenvalue* dan *eigenvector* dari set *database* wajah. Untuk dapat melakukan rutin perintah tersebut dibutuhkan *library* python sebagai berikut:

```
import Image
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
```

List 1. Proses import library pada rutin Python

Perhatikan List dibawah untuk kode pemrosesan dari file gambar ke dalam bentuk *array numpy* program yang mana nantinya set *array*:

```
gbr1 =
"D:\\FaceReconictionProject\\img\\at\\tr
ain01\\1.pgm"
gbr2 =
"D:\\FaceReconictionProject\\img\\at\\tr
ain01\\2.pgm"
gbr3 =
"D:\\FaceReconictionProject\\img\\at\\tr
ain01\\3.pgm"
# load train image ke matrix X, dan y
untuk index gambar
print "#1 - Load gambar dari file ->
array matrix"
print "gambar1:"
print gbr1
im = Image.open(gbr1)
im = im.convert("L")
X.append(np.asarray(im,
dtype=np.uint8))
y.append(1)
```

```
print "gambar2:"
print gbr2
im = Image.open(gbr2)
im = im.convert("L")
X.append(np.asarray(im,
dtype=np.uint8))
y.append(2)
print "gambar3:"
print gbr3
im = Image.open(gbr3)
im = im.convert("L")
X.append(np.asarray(im,
dtype=np.uint8))
y.append(3)
```

List 2. Load gambar ke dalam *numpy array*

File 1.pgm,2.pgm dan 3.pgm merupakan set gambar database wajah yang dibaca dengan perintah kode diatas, untuk melihat gambar plotnya dapat dibentuk dengan baris kode berikut:

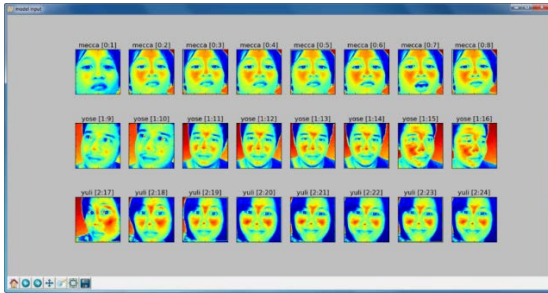
```
model = plt.figure("Model Input")
# judul
model.text(.5,.95,'Model',horizontalalignment='center')
for i in xrange(len(X)):
    ax0 = model.add_subplot(1,3,(i+1))
    plt.setp(ax0.get_xticklabels(), visible=False)
    plt.setp(ax0.get_yticklabels(), visible=False)
    plt.title("%s #s" % ("model",str(i+1)))
    plt.imshow(np.asarray(X[i], cm.gray))
model.show ()
print "-----"
```

List 3. Plot gambar yang dibaca dari database train

PENGENALAN POLA WAJAH

Pendeteksi pola wajah dapat dilakukan dengan deretan kode program dengan metode cascade classification dengan bantuan library OpenCV, objek wajah yang ditemukan yang dapat mengekstrak. Proses yang lebih sulit adalah mengetahui objek wajah tersebut mirip dengan objek yang mana yang sudah ada dari database.

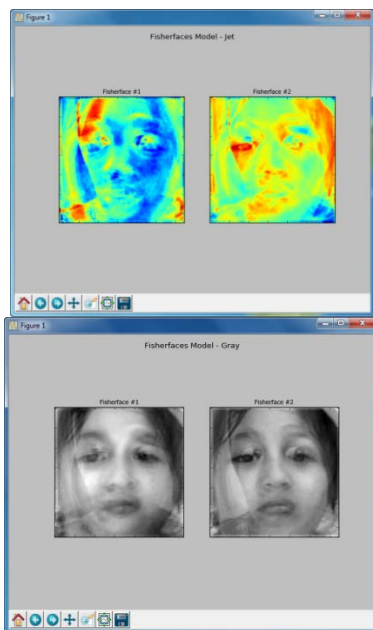
Gambar akan disusun dalam satu baris kolom untuk masing-masing gambar disimpan dalam variable X, dan label disimpan dalam matriks y.



Gambar 1. Plot model input

Model berikutnya diproses dengan algoritma PCA sehingga didapat eigenvalue dan eigenvectornya, kemudian output dari PCA diproses dengan algoritma LDA.

Program akan menghitung *eigenvector* baru dengan rumus $pca_eigenvectors \cdot lda_eigenvectors$, berikut plot gambar untuk *Fisherface*-nya:



Gambar 2. Plot Fishierface untuk model

LCD RPI (A) yang digunakan berukuran 3,5 inci dari Waveshare dengan dilengkapi touchscreen, untuk dapat menggunakannya download image Rasbian dan driver dari Waveshare di http://www.waveshare.com/wiki/3.5inch_RPi_LCD_%28A%29 kemudian di write ke MicroSD. Saat booting pertama kali Raspberry Pi sudah dapat menampilkan dengan LCD.

Karena ukuran LCD lebih kecil (3,5”) perlu di-resize ukuran frame image yang dicapture dari kamera agar muat dilayar LCD, peneliti mengecilkan sepertiga dari ukuran normal. Berikut hasil pendeteksi dan pengenalan pola wajah menggunakan LCD display.



Gambar 3 Tampilan LCD 3,5”

Pengenalan dengan dua objek wajah dapat dilakukan dengan program yang jalan di Raspberry Pi, pengenalan berjalan dengan baik dan dapat membedakan dengan benar dua wajah yang berbeda.



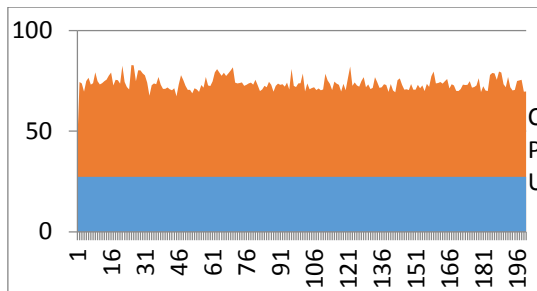
Gambar 4. Pengenalan dua objek wajah di Raspberri Pi

AKURASI DAN PERFORMA

Berdasarkan data pengujian ini didapat akurasi $129/132 \cdot 100 = 97,73 \%$. Proses pengujian ini menggunakan dua sampel objek wajah “yose” dan “yuliani” dengan ada tiga wajah yang dikenali dalam database dengan 200 sampel output deteksi. Berdasarkan data Tabel output sapling output yang benar 168 dan yang salah 32 output, berdasarkan data pengujian ini

didapat kaurasi $168/200 * 100 = 84 \%$. Kondisi input wajah tidak statistic, tetapi berubah-ubah dengan beberapa bentuk mimic wajah.

Berdasarkan data hasil pengujian ini dapat diplot grafik performance terhadap jumlah proses data terhadap performance CPU.



Grafik 5. Persentase penggunaan CPU Raspberry Pi pengelan 1 objek wajah

Berdasarkan data didapat rata-rata penggunaan CPU sebesar 73.55 % saat proses program dibandingkan dengan saat startup program 27.3% penggunaan resource CPU saat proses program naik sebesar 46.25 %.

DAFTAR PUSTAKA

- [1] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. *Computer Vision - ECCV 2004* (2004), 469-481.
- [2] A.K.Jain, S.J.R. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 3 (1991),252-264.
- [3] Belhumeur, P. N., Hespanha, J., and Kriegman, D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 7 (1997), 711-720.
- [4] Brunelli, R., and Poggio, T. Face recognition through geometrical features. In *European Conference on Computer Vision (ECCV)* (1992), pp. 792-800.
- [5] Cardinaux, F., Sanderson, C., and Bengio, S. User authentication via adapted statistical models of face images. *IEEE Transactions on Signal Processing* 54 (January 2006), 361-373.
- [6] Chiara Turati, Viola Macchi Cassia, F. S., and Leo, I. Newborns face recognition: Role of inner and outer facial features. *Child Development* 77, 2 (2006), 297-311.
- [7] Duda, R.O., Hart, P. E., and Stork, D. G. *Pattern Classification* (2nd Edition), 2 ed. November 2001.
- [8] Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals Eugen.* 7 (1936), 179-188.
- [9] Kanade, T. *Picture processing system by computer complex and recognition of human faces*. PhD thesis, Kyoto University, November 1973.
- [10] Lee, K.-C., Ho, J., and Kriegman, D. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 27, 5 (2005).
- [11] Maturana, D., Mery, D., and Soto, A. Face recognition with local binary patterns, spatial pyramid histograms and naive bayes nearest neighbor classification. *2009 International Conference of the Chilean Computer Science Society (SCCC)* (2009), 125-132.
- [12] Rodriguez, Y. *Face Detection and Verification using Local Binary Patterns*. PhD thesis, Ecole Polytechnique Federale De Lausanne, October 2006.
- [13] Turk, M., and Pentland, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3 (1991), 71-86.
- [14] Wiskott, L., Fellous, J.-M., Kruger, N., and Malsburg, C. V. D. Face recognition by elastic bunch graph matching. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 19 (1997), 775-779.
- [15] Zhao, W., Chellappa, R., Phillips, P., and Rosenfeld, A. Face recognition:

- A literature survey. *Acm Computing Surveys (CSUR)* 35, 4 (2003), 399-458.
- [16] Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE CVPR*, 2001
- [17] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. *IEEE ICIP* 2002, Vol. 1, pp. 900-903, Sep. 2002.