

Aplikasi *Principle Component Analysis* (PCA) Untuk Mempercepat Proses Pendeteksian Obyek Pada Sebuah Image

Liliana

Jurusan Teknik Informatika, Fakultas Teknologi Industri Universitas Kristen Petra,
Surabaya

Abstrak

Mendeteksi sebuah obyek pada sebuah image merupakan aplikasi umum dalam komputer vision. Alat-alat bantu keamanan sangat membutuhkan aplikasi komputer vision, terutama untuk diaplikasikan secara real time. Kendala utama yang sering digunakan adalah metode yang menghasilkan perhitungan yang akurat membutuhkan waktu proses yang cukup besar. Oleh karena itu diperlukan sebuah metode pencocokan image yang cepat.

Pencocokan image tanpa metode tertentu akan membutuhkan waktu semakin besar seiring dengan besarnya ukuran image yang dibandingkan. Hal ini disebabkan oleh cara perbandingannya, yaitu setiap pixel dibandingkan satu per satu. PCA adalah salah satu metode yang digunakan untuk memperkecil dimensi dari sekumpulan data. Dengan diaplikasikannya PCA pada proses pencocokan image, maka ukuran image yang besar bukan lagi menjadi faktor perbandingan. PCA akan menghasilkan komponen-komponen perbandingan baru, yang signifikan sebagai perbandingan dan dalam jumlah yang jauh lebih sedikit.

Dari hasil uji coba, dapat disimpulkan bahwa dengan data yang homogen, maka pencocokan dengan menggunakan PCA akan lebih akurat dibandingkan jika menggunakan data yang lebih bervariasi. Waktu pencocokan hanya terpaut sedikit karena ukuran image yang tidak terlalu besar. Jika ukuran image input semakin besar, maka waktu yang dibutuhkan untuk proses pencocokan juga akan semakin meningkat.

Kata kunci: PCA, deteksi obyek, komputer vision.

Latar Belakang

Dengan perkembangan teknologi kamera dan juga perkembangan aplikasi komputer vision, maka metode pendeteksian sebuah obyek dalam suatu gambar juga semakin berkembang [1]-[11]. Pendeteksian obyek ini kemudian akan dimanfaatkan untuk pengenalan obyek yang dideteksi tersebut atau memantau adanya gerakan dari obyek tersebut. Dalam perkembangan selanjutnya, pengenalan obyek digunakan untuk mendeteksi keberadaan dari sebuah obyek yang di jadikan target

pencarian, atau mengenali jenis dari obyek yang ada pada sebuah image. Sedangkan pemantauan adanya gerakan dari obyek yang dijadikan target diaplikasikan pada perangkat – perangkat lunak untuk bagian keamanan; pengontrolan terhadap mesin – mesin produksi barang, terutama jika terjadi gerakan di luar jalur yang seharusnya atau adanya pemindahan posisi barang yang tidak dikehendaki.

Perangkat – perangkat lunak yang membutuhkan pendeteksian adanya gerakan seringkali diaplikasikan secara

real time [10], [11]. Oleh karena itu, aplikasi-aplikasi demikian memerlukan sebuah metode yang proses pengerjaannya tidak membutuhkan waktu yang lama dan hasilnya akurat. Sedangkan aplikasi-aplikasi yang bertujuan untuk mengenali obyek yang terdapat dalam sebuah image membutuhkan banyak image lain sebagai referensi [1]–[10]. Image-image referensi tersebut sudah disimpan terlebih dahulu ke dalam database. Setelah itu, dilakukan proses pencocokan untuk mencari kesesuaian image input (target image) dengan image-image referensi.

Lama proses pencarian tersebut dipengaruhi oleh banyaknya data, kompleksitas data dan ukuran dari data image itu sendiri. Kondisi yang paling ideal adalah dengan menggunakan data image sebagai referensi dalam jumlah minimal dan ukuran yang sekecil mungkin, dan sesedikit mungkin *noise* (adanya obyek lain selain obyek yang ingin dideteksi), dapat menghasilkan output yang akurat. Sementara, kesulitan yang dihadapi untuk mencapai kondisi ideal tersebut adalah, keakuratan hasil yang sulit didapat jika image referensi terlalu sedikit, kurang jelasnya bentuk obyek yang ingin dideteksi jika resolusi gambar/image rendah dan metode tersendiri untuk menghilangkan noise.

Faktor yang paling mempengaruhi lama proses pencocokan image adalah jumlah image referensi. Jumlah image yang banyak bukan berarti peningkatan akurasi karena yang seringkali terjadi adalah adanya redundansi image atau image yang berbeda namun memberikan informasi yang sama. Image-image semacam ini hanya perlu disimpan salah satunya saja, untuk digunakan sebagai referensi. Untuk memfilter image-image yang

mempunyai redundansi yang tinggi, salah satu metode yang dapat dipakai adalah Principle Component analysis (PCA).

PCA merupakan sebuah metode dalam aljabar linear yang bertujuan untuk mengubah basis dari sekumpulan data sedemikian rupa sehingga basis yang baru akan menampakkan signifikansi dari data-data yang ada. PCA dapat digunakan untuk mempercepat proses pencarian karena PCA mampu mengubah data-data awal yang berdimensi banyak menjadi data-data baru yang berdimensi lebih kecil.

PCA

Metode PCA memperlakukan data sebagai vektor dari ruang n -dimensi [1], [2]. Dengan sekumpulan data berupa vektor sebagai input, maka masing-masing data direpresentasikan dalam n variable berupa nilai skalar sebagai hasil proyeksi data tersebut ke setiap basis dari n -dimensi. Tujuan dari PCA adalah mencari basis yang baru untuk merepresentasikan ulang data tersebut ke dalam m variable hasil proyeksi data ke m -dimensi. Dimana m lebih kecil dari n . Dengan berkurangnya jumlah dimensi dari data input, maka proses pencocokan data akan berkurang dari n kali setiap pencocokan menjadi m kali. Banyaknya dimensi data ditentukan oleh besarnya resolusi image. Hal ini menyebabkan dimensi data menjadi sangat besar. Semakin besarnya dimensi data menyebabkan waktu pencocokan per data semakin besar juga.

Langkah pertama untuk mengurangi redundansi data adalah mengindikasikan data-data yang mempunyai korelasi tinggi [2]. Untuk mengetahui korelasi data, dapat digunakan operasi *dot product*. Dari operasi dot product, dapat diketahui besarnya sudut antara dua buah vektor. Dan jika sudut yang

terbentuk di antara dua buah vektor mendekati nol, maka kedua vektor tersebut mendekati paralel, yang berarti adanya korelasi tinggi di antara kedua buah vektor tersebut. Karena operasi dilakukan pada vektor, maka data berupa image (I) harus diubah terlebih dahulu ke bentuk vektor kolom (X) [1], [2]. Jika data berupa image dengan ukuran $m \times n$, dimana m dan n adalah ukuran tinggi (arah vertikal) dan lebar (arah horisontal) dari image tersebut, maka untuk menjadikannya vektor dapat digunakan persamaan 1.

$$x_{(i)} = I_{(i/n, i \% n)} \quad (1)$$

"/" adalah operasi pembagian bulat, sedangkan "%" adalah sisa bagi dari pembagian bulat. i menunjukkan komponen ke- i dari vektor X , yang bernilai $[0, (mn - 1)]$.

Setelah semua data image sebanyak k diubah menjadi vektor kolom, kemudian digabungkan menjadi sebuah matriks (X) dengan masing-masing vektor sebagai kolom dalam matriks X tersebut. Sehingga matriks X akan berukuran $mn \times k$.

Jika terdapat sebuah matrix Y , dimana setiap baris dari matriks Y merepresentasikan sebuah data, maka *covariant matrix* (C) dari matriks Y didapat dengan menggunakan persamaan 2 [2].

$$C_y \equiv \frac{1}{n-1} YY^T \quad (2)$$

Jika masing-masing data tidak berkorelasi satu sama lain, maka matriks C hanya akan mempunyai nilai yang besar pada diagonalnya saja, sedangkan elemen yang lain akan bernilai kecil, mendekati nol. Dan ketidakkorelasi yang sempurna akan membuat matriks C menjadi matriks diagonal. Dengan asumsi awal bahwa sekumpulan data yang dimiliki adalah data-data dengan

korelasi tinggi, maka diperlukan sebuah basis baru yang ortogonal (P) yang dapat mengubah X sedemikian rupa sehingga menjadi matriks Y , dimana covariance matriks dari matriks Y adalah matrix diagonal. Hubungan antara matriks X , Y dan basis yang baru adalah sebagai berikut:

$$Y = P X \quad (3)$$

Dimana Y merupakan koordinat baru dari X terhadap basis yang baru, P . Dengan demikian, maka yang perlu diperhatikan adalah bagaimana mendapatkan basis yang benar, yang membuat data dalam matriks Y tidak berkorelasi satu sama yang lain. Karena matriks Y tidak diketahui pada awalnya, maka dilakukan substitusi persamaan 3 ke persamaan 1, seperti yang dijelaskan dalam penjabaran di bawah ini [2].

$$\begin{aligned} C_Y &= \frac{1}{n-1} YY^T \\ &= \frac{1}{n-1} (PX)(PX)^T \\ &= \frac{1}{n-1} PXX^T P^T \\ &= \frac{1}{n-1} P(XX^T)P^T \\ C_Y &= \frac{1}{n-1} PAP^T \end{aligned} \quad (4)$$

Dengan memperhatikan A sebagai XX^T , maka A merupakan sebuah matriks simetris. Sebuah matriks simetri akan didiagonalisasi oleh sebuah matriks ortogonal yang terbentuk dari eigen-eigen vektornya, dengan menggunakan persamaan 4 di bawah ini.

$$A = EDE^{-1} \quad (5)$$

Dimana E adalah eigen vektor dari A dan D adalah matriks diagonal yang berisi nilai eigen dari A yang berkoresponden dengan eigenvektor tersebut. Karena invers matriks dari eigenvektor sebuah matriks yang simetri adalah transpose dari eigenvektor itu, maka persamaan 5 dapat diubah menjadi persamaan 6.

$$A = EDE^T \quad (6)$$

Dari penjabaran persamaan 4, maka A dapat dituliskan seperti persamaan 7.

$$A = P^T \frac{n-1}{1} C_y P \quad (7)$$

Persamaan 6 dan 7 mempunyai persamaan bentuk. Dengan asumsi bahwa $P^T = E$ and $(n-1)/1 C_y = D$. untuk membuktikan bahwa asumsi tersebut benar, maka dapat dilihat penjabaran dibawah ini. Pada penjabaran rumus di bawah ini, A akan disubsitusi dengan P^TDP [2].

$$\begin{aligned} C_Y &= \frac{1}{n-1} P A P^T \\ &= \frac{1}{n-1} P (P^T D P) P^T \\ &= \frac{1}{n-1} (P P^T) D (P P^T) \\ &= \frac{1}{n-1} (P P^{-1}) D (P P^{-1}) \\ C_Y &= \frac{1}{n-1} D \end{aligned} \quad (8)$$

Dengan demikian, terbukti bahwa $D = (n-1)/1 C_y$. Dengan pembuktian tersebut, maka basis baru yang ortogonal bisa didapatkan dari eigenvektor dari matriks A, dimana A adalah perkalian dari matri data awal, X dengan transpose matriksnya sendiri.

Dari eigenvektor yang dihasilkan, hanya perlu diambil yang mempunyai nilai eigen yang signifikan saja. Dengan demikian, maka koordinat yang baru adalah hasil proyeksi data terhadap eigenvektor yang signifikan tersebut saja. Hal ini juga berarti jumlah komponen koordinat yang baru akan jauh lebih sedikit dibandingkan dengan data asalnya.

Algoritma

Keseluruhan proses untuk mendeteksi sebuah obyek yang terdapat

dalam suatu image dapat dibagi menjadi dua bagian. Pertama mencari basis baru sesuai dengan image-image yang dijadikan referensi. Proses yang kedua adalah mencari posisi obyek target di dalam image yang dijadikan inputan. Kedua proses tersebut dapat dituliskan urutan kerjanya sebagai berikut.

Proses mencari basis yang baru:

- Mengubah semua image referensi menjadi vektor kolom
- Menggabungkan semua vektor kolom tersebut menjadi sebuah matrix data (X).
- Mengalikan matriks X dengan transposenya, sehingga dihasilkan matriks simetri (A).
- Mencari eigenvektor dari matriks A dan menggunakannya sebagai basis yang baru.
- Normalkan eigenvektor tersebut.
- Menghitung koordinat yang baru dari setiap image referensi.

Proses mencari lokasi dari obyek yang ditargetkan dari sebuah image inputan:

- Mengubah image input menjadi vektor kolom
- Mengubah vektor kolom tersebut menjadi koordinat yang baru (vektor kolom yang baru) dengan menggunakan eigenvektor yang dihasilkan pada proses sebelumnya.
- Melakukan proses pencocokan dengan image-image referensi yang sudah di jadikan koordinat baru pada proses sebelumnya. Jika besarnya perbedaan antara koordinat baru dari image input dengan image referensi lebih kecil dari threshold yang sudah ditentukan sebelumnya maka image input tersebut mempunyai klasifikasi yang sama dengan image referensi.
- Jika image input mempunyai ukuran yang lebih besar dari image referensi, maka proses pencocokan dilakukan

pada setiap sub image input, sesuai dengan ukuran image referensi.

Uji Coba

Untuk melakukan pengujian terhadap perangkat lunak yang diimplementasikan, disiapkan sembilan image yang berbeda berupa foto wajah, seperti tampak pada gambar 1. Masing-masing foto berukuran 92 x 112. Obyek yang ditargetkan adalah mulut. Sebagai image referensi, bagian mulut dari semua foto tersebut diambil secara manual. Keseluruhan foto bagian mulut saja, dapat dilihat pada gambar 2. Bagian mulut berukuran 36 x 20. Untuk proses pertama, maka menggunakan hanya image bagian mulut saja. Sedangkan untuk proses yang kedua, digunakan foto wajah keseluruhannya.

Dari eigen vector yang dihasilkan oleh proses pertama, didapatkan 9 nilai eigen yang signifikan (tidak bernilai nol). Hal ini berarti koordinat yang baru terletak pada ruang 9 dimensi.

Selain pengujian dengan menggunakan PCA, sebagai pembanding untuk menguji kinerja dari PCA itu, digunakan pencarian lokasi "mulut" pada image input dengan cara membandingkan setiap pixel satu persatu.

Proses pencocokan dilakukan dengan mengambil sebagian dari image input, dilanjutkan dengan proses dua. Setelah mendapatkan koordinat yang baru dari sebagian image input tersebut, maka koordinat tersebut di cocokkan dengan koordinat-koordinat baru dari image referensi, jika perbedaan antara kedua koordinat itu lebih kecil dari threshold, maka bagian yang sedang dicek tersebut adalah mulut. Untuk menandai area mulut yang dideteksi, maka digunakan kotak putih. Setelah melakukan pengujian terhadap

kesembilan foto wajah tersebut, maka hasilnya dapat dilihat pada tabel 1.



gambar 1. Foto wajah yang digunakan sebagai input saat proses pencocokan.



Gambar 2. Gambar foto bagian mulut yang digunakan sebagai referensi.

Table 1. hasil uji coba dengan menggunakan 9 foto sebagai input.

Perbandingan pixel per pixel	Perbandingan dengan menggunakan PCA
	

(84 , 51)	(85 , 48)
	
(86 , 51)	(86 , 52)
	
(88 , 48)	(88 , 48)
	
(85 , 50)	(84 , 51)
	
(85 , 48)	(85 , 48)
	
(87 , 49)	(87 , 49)
	
(88 , 44)	(87 , 48)

	
(88 , 46)	(88 , 46)
	
(83 , 50)	(84 , 49)

Angka yang tertulis pada tabel 1 adalah titik pusat dari kotak berwarna putih. Dari tabel satu, dapat diketahui bahwa pendeteksian obyek “mulut” dikenali pada posisi yang hampir sama dengan yang dikenali dari pencarian dengan membandingkan pixel satu demi satu.

Sedangkan untuk waktu yang diperlukan untuk melakukan proses pencarian adalah 7 detik jika menggunakan perbandingan pixel demi pixel. Sedangkan waktu yang dibutuhkan jika menggunakan PCA adalah 5 detik.

Kesimpulan

Metode PCA dapat diaplikasikan untuk mendeteksi sebuah obyek pada image input. Mengenai percepatan proses pencocokan, belum terlihat jelas pada pengujian sistem yang telah dilakukan saat ini. Untuk mendapatkan hasil yang lebih maksimal, maka kedepan perlu diadakan uji coba sistem untuk image input yang lebih banyak dan lebih besar ukurannya.

Untuk kinerja PCA, juga belum terlihat maksimal. Beberapa kemungkinan yang terjadi adalah ketidakhomogenan dalam image

referensi. Karena pada dasarnya PCA adalah sebuah metode yang membedakan obyek berdasarkan intensitas, bukan berdasarkan bentuk morfologinya, maka untuk pengujian perlu diuji juga untuk image yang lebih homogen.

Daftar Pustaka

- [1] M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
- [2] J. Shlens, "A Tutorial on Principal Component Analysis", 2005.
- [3] X. Li, Y. He, "A Novel Approach to Pattern Recognition Based on PCA-ANN in Spectroscopy", *ADMA 2006*, LNAI 4093, pp. 525-532, 2006.
- [4] L. Malagon-Borja, O. Fuentes, "Object detection using image reconstruction with PCA", *Image Vis. Comput*, 2007, doi:10.1016/j.imavis.2007.03.004
- [5] R.P.W. Duin, M. Loog, R. Haeb-Umbach, "Multi-class Linear Feature Extraction by Nonlinear PCA", *icpr*, vol. 2, pp. 2398, 15th International Conference on Pattern Recognition (ICPR'00), 2000.
- [6] A.F. Abate, M. Nappi, D. Riccio, G. Sabatino, "2D and 3D face recognition: A survey", *Pattern Recognition Letters*, vol. 28, pp. 1885-1906, January 2007.
- [7] Neerja, E. Walia, "Face Recognition Using Improved Fast PCA Algorithm", *Congress on Image and Signal Processing proceedings*, pp. 554-558, 2008.
- [8] J. Krueger, D. Kochelek, M. Robinson, M. Escarra, "Face Detection Using Eigenfaces", *The Connexions Project* and licensed under the Creative Commons Attribution License, December 2004.
- [9] W.S. Yambor, B.A. Draper, J.R. Beveridge, "Analyzing PCA-based Face Recognition Algorithm: Eigenvector Selection and Distance Measures", *World Scientific Press*, Singapore, 2001.
- [10] H. Bae, S-S. Kim, "Real-time face detection and recognition using hybrid-information extracted from face space and facial features", *Image and Vision Computing*, vol. 23, pp. 1181-1191, 2005.
- [11] E. Truco, A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, New Jersey, 1998.