ISSN : 2252-3839 (Print)
ISSN : 2549-2403 (On Line)
DOI : 10.28989/compiler.v8i2.538
http://ejournals.stta.ac.id/index.php/compiler/

ANALYSIS OF COMBINATION KNOWLEDGE ACQUISITION OF *HAAR TRAINING* FOR OBJECT DETECTION ON THE VIOLA JONES METHOD

Haruno Sajati[1)], Anggraini Kusumaningrum[2)], Nur Hanifah[3)]
[1,2,3]Departemen Informatika, Sekolah Tinggi Teknologi Adisutjipto
Jl. Janti, Blok-R Lanud Adisutjipto Yogyakarta
Email : [1]harunosajati@stta.ac.id, [2]anggraini@stta.ac.id, [3]hanifsay837@gmail.com

Abstract

Viola Jones method uses the file classifier to object detection. The training process to create object classifier file requires very high computer resources and time which is directly proportional to the amount of training data. The amount of training data determines the accuracy of object detection. The long training process is caused because the computer has low specifications and the distribution of Haar Training files will speed up the process of vector file formation, minimize errors when cutting Haar features on positive objects and also minimize errors that occur during the training process. The problems that arise next are how to overcome this so that a better knowledge is obtained. This study provides analysis results of the process of merging knowledge acquisition and its effect on the accuracy of object detection using the Viola-Jones method with the final result undetected object decrease 52.62% *and object detected increase 23.78*%.

Keywords*: Knowledge Acquisition, Training, Object Detection, Viola-Jones*

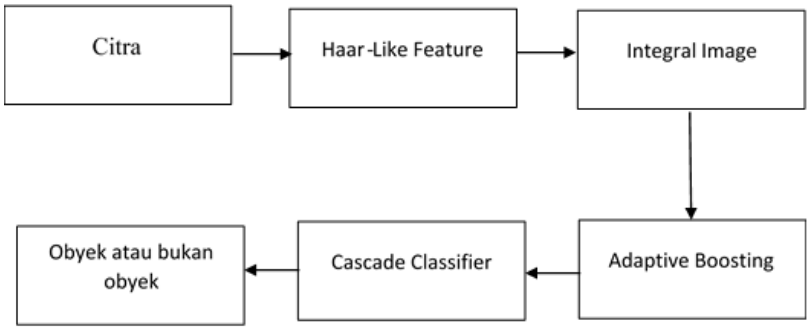
1. Latar Belakang Masalah

Metode Viola Jones merupakan algoritma yang banyak digunakan untuk melakukan pendeteksian suatu obyek. *Haar Training* memerlukan dua tipe gambar obyek dalam proses *training* yang dilakukan, yaitu *positive samples* yang berisi gambar obyek dan *negative samples* berisi gambar selain obyek yang ingin dikenali, umumnya berupa gambar *background* (tembok, pemandangan, lantai dan gambar lainnya). Saat ini banyak yang sudah melakukan penelitian tentang pendeteksi suatu obyek yang bertujuan untuk mengetahui ada atau tidaknya obyek pada suatu gambar dengan menggunakan Metode Viola Jones tersebut [1][2][3][4][5][6], namun dalam kondisi tertentu *Haar-cascade* mendeteksi obyek palsu sehingga akan memunculkan beberapa kandidat obyek. Obyek palsu ini muncul karena kurangnya data *training*, penambahan data *training* akan mengakibatkan lamanya proses *Haar Training*, memperlama proses pemotongan fitur Haar pada obyek positif dan juga rentan terhadap kesalahan saat melakukan pemotongan fitur *Haar* pada obyek positif [4]. Proses *training* yang lama ini diakibatkan karena komputer mempunyai spesifikasi yang rendah dan dengan pembagian file *training* akan mempercepat proses pembentukan file *vector*, meminimalisir kesalahan saat melakukan pemotongan fitur *Haar* pada obyek positif dan juga meminimalisir kesalahan yang terjadi saat melakukan proses *training* [5][6][13][14]. Permasalahan yang timbul kemudian adalah bagaimana cara penggabungan *knowledge acquisition* untuk menurunkan kesalahan dalam pendeteksian obyek dan pengaruh dari proses penggabungan *knowledge acquisition* terhadap akurasi pendeteksian obyek[9][10][11][12][13].

2. Metodologi Penelitian

2.1 Metode Viola – Jones

Dalam *framework* Viola-Jones disediakan banyak *library* untuk melakukan proses seleksi fitur, fitur yang merupakan fungsi dasar untuk meningkatkan seleksi dikenal sebagai *Haar-Like feature*. Viola-Jones memperkenalkan sebuah representasi *image* baru yang dikenal sebagai *Integral Image* yang tidak terpisahkan. Dengan metode ini *Haar-Like features* dapat dihitung pada setiap skala atau lokasi dalam waktu yang konstan. Metode Viola-Jones menggabungkan empat kunci utama yaitu *Haar-Like Feature*, *Integral Image*, *AdaBoost* dan *Cascade classifier*. Gambar 1 merupakan gambar Skema Proses Deteksi Obyek dengan Metode Viola-Jones.



Gambar 1. Skema Proses Deteksi Obyek dengan Metode Viola-Jones.

2.2 Kebutuhan Data

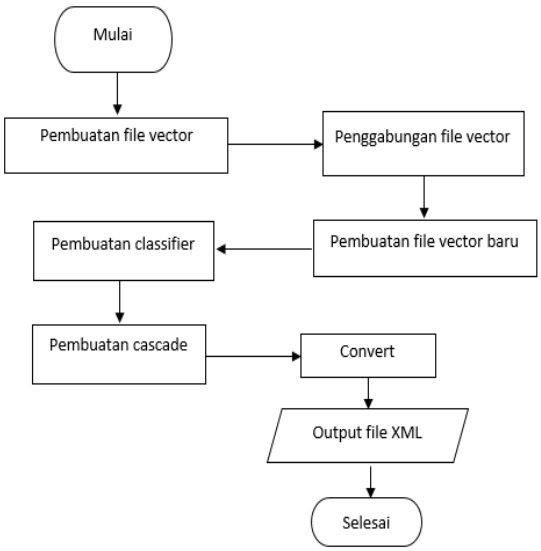
Data yang digunakan terbagi dalam 2 bagian yaitu citra positif dan citra negatif. Citra sampel positif yaitu citra sampel yang di dalamnya terdapat objek (*flash disk*) sebanyak 400 sampel. Sedangkan Citra sampel negatif adalah citra sampel yang di dalamnya tidak terdapat objek yang akan dideteksi dan diambil secara acak sebanyak 1000 sampel. Pemilihan flash disk sebagai obyek penelitian adalah jumlah varian dari *flash disk* cukup banyak dan keberadaannya mudah ditemukan. Flash disk juga memiliki dimensi fisik yang cukup besar untuk dipindai dalam Region of Interest (ROI) saat pendeteksian obyek.

2.3 Kebutuhan File Pustaka

File pustaka yang digunakan pada penelitian ini adalah OpenCV. Open Source Computer Vision Library merupakan software pustaka yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV dirilis di bawah lisensi permisif BSD. Lisensi BSD adalah keluarga dari lisensi perangkat lunak bebas yang permisif dan memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu mengungkapkan kode sumbernya. OpenCV juga memiliki antar muka yang mendukung bahasa pemrograman C++ [15].

2.4 Perancangan Sistem

Gambar 2 merupakan perancangan alur sistem yang digunakan dalam perangkat lunak dari sebuah gambar yang akan diproses untuk di filter menjadi sebuah data output yang berupa file dalam format XML



Gambar 2. Proses penggabungan *knowledge* untuk pendeteksian obyek

. Untuk mendapatkan file XML harus terlebih dahulu mulai pembuatan file *vector* dengan menggunakan perintah *create sample command* yang di dalamnya sudah berisi citra positif. Kemudian membuat file *vector* lain dengan data berbeda yang nantinya akan dilakukan penggabungan *vector*. Untuk melakukan penggabungan *vector* akan menghasilkan *vector* baru hasil dari penggabungan tersebut. Untuk pembuatan *classifier* nya terlebih dahulu melakukan *Haar Training* yang di dalamnya terdapat citra negatif dan file *vector*. File *vector* hasil penggabungan itulah yang akan digunakan pada proses *Haar Training*. Proses ini akan membentuk direktori *classifier*. Kemudian pembentukan *cascade* dengan cara melakukan perpindahan isi direktori *classifier* ke dalam direktori *cascade*. Tahap berikutnya adalah pembentukan file XML melalui proses *convert* untuk menggabungkan data tersebut menjadi satu file *classifier* dengan nama *haarcascade*.xml.

2.5 Perancangan Pembuatan Haar Like Feature

Dalam proses penggabungan vector dibutuhkan beberapa File vector yaitu:

1. Sample.vec
   File sample.vec berisikan code yang akan dikenali pada *library* OpenCV untuk mendeteksi kumpulan *flash disk* kelompok pertama.
2. Sample1.vec
   File sample.vec berisikan code yang akan dikenali pada *library* OpenCV untuk mendeteksi kumpulan *flash disk* kelompok kedua.
3. Gabungan.vec
   File gabungan.vec berisikan code yang akan dikenali pada *library* OpenCV untuk mendeteksi kumpulan *flash disk* kelompok gabungan.

2.6 Proses Pembuatan *Cascade*

Tahap pembuatan *cascade* dilakukan melalui beberapa tahap seperti pembuatan *list negatif, pemberian fullpath* gambar negatif, pembentukan *file* negNew.txt, persiapan *file* positif, pembuatan *object info*, terbentuk *file* info.txt, *full path* untuk *file* info.txt (*rawdata*), pembuatan *file vector*, terbentuk *file vector*, pembuatan *classifier*, terbentuk *training cascade*, penyatuan *file cascade* menjadi *cascade classifier* lengkap sampai pada akhirnya keluarlah *output* berupa file XML. Proses tersebut dilakukan pada jumlah data

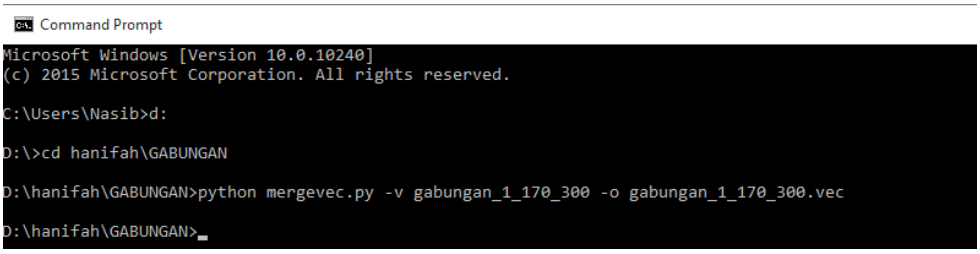
gabungan sebanyak 11 kali percobaan pendeteksian yang memiliki jumlah *flash disk* berbeda. Pada proses penggabungan ini akan diambil file *vector* nya saja, file *vector* tersebut yang bernama sample.vec. File sample.vec ini nantinya merupakan penggabungan antara sample.vec1 dan sample.vec 2.

2.7 Membuat Folder Gabungan

Setelah mendapatkan file *vector* dari beberapa sample data maka selanjutnya dilakukan penggabungan file *vector* tersebut ke dalam satu folder

2.8 Membuat File Gabungan Vector

Setelah file *vector* yang akan di-*combine* dikumpulkan ke dalam satu folder, maka tahap selanjutnya adalah menggabungkan 2 *vector* tersebut ke dalam 1 folder yang nantinya akan menghasilkan 1 *vector*. Proses melakukan gabungan *vector* ditunjukkan pada Gambar 4.



Gambar 4. Proses Penggabungan Vector

Setelah pembuatan *vector* selesai dilakukan, selanjutnya adalah proses merubah *vector* yang sebelum digabung menjadi *vector* yang sudah digabung dan mengubah jumlah gambar positif dan gambar negatif sesuai dengan jumlah yang akan dilakukan proses *haar-training*. Proses ini akan menghasilkan output XML. File ini yang akan digunakan untuk proses pendeteksian obyek pada tahap berikutnya.

3. Hasil dan Pembahasan

3.1 Implementasi Hasil

Gambar 6 menunjukkan contoh dan proses perhitungan hasil pendeteksian obyek menggunakan metode Viola Jones.

Gambar 6. Hasil Sebelum Penggabungan pada Data Pertama

Dapat dilihat dari gambar 6 yang berhasil terdeteksi 4 *flash disk* dimana *flash disk* lebih mendominasi berada di area lingkaran dan yang tidak terdeteksi ada 5 *flash disk*.

Gambar 7. Hasil Sebelum Penggabungan pada Data Kedua

Dapat dilihat dari gambar 7 yang berhasil terdeteksi 5 *flash disk* dimana *flash disk* lebih mendominasi berada di area lingkaran dan yang tidak terdeteksi ada 5 *flash disk*.

Gambar 8. Hasil Penggabungan

Gambar 8 merupakan hasil pendeteksian menggunakan file classifier yang diperoleh dari penggabungan 2 *vector*. Hasil percobaan pendeteksian pada Gambar 8 menunjukkan hasil sebanyak enam *flash disk* dimana *flash disk* lebih mendominasi berada di area lingkaran dan yang tidak terdeteksi ada 3 *flash disk*.

3.2 Analisa hasil Pendeteksian Obyek

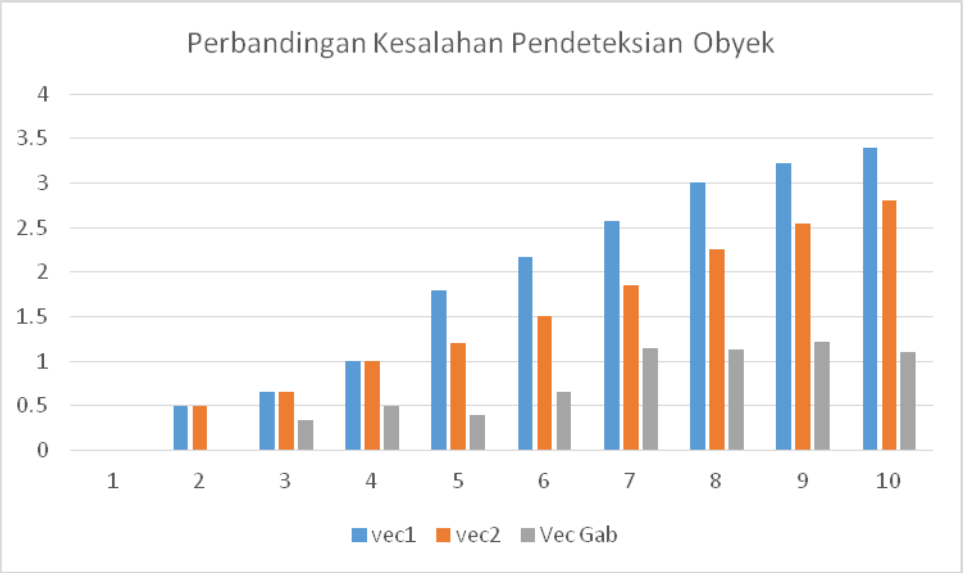
Pada penelitian ini dilakukan pengujian terhadap 10 obyek *flash disk* dengan jumlah yang berbeda. Perhitungan means dilakukan dengan menjumlahkan data obyek yang tidak terdeteksi/terdeteksi di bagi jumlah *flash disk*, lalu hasilnya saling dijumlahkan. Hasil penjumlahan means pada data percobaan Vec 1 dan data percobaan Vec 2 pada Tabel 1 dijumlahkan kembali dan dibagi 10. Untuk mengetahui hasil prosentase dapat dihitung dengan cara data gabungan dibagi hasil penjumlahan data 1 dan data 2 lalu dikalikan 100%.

Table 1. Vec Gabungan: N-Pos 40 N-Neg 40

Jumlah *Flash disk*	Vec 1: N-Pos 20 N-Neg 40 (Means)	Vec 2: N-Pos 20 N-Neg 40 (Means)	Vec Gabungan: N-Pos 40 N-Neg 40 (Means)
1	0	0	0
2	0.5	0.5	0
3	0.66	0.66	0.33
4	1	1	0.5
5	1.8	1.2	0.4
6	2.166	1.5	0.66
7	2.571	1.85	1.142
8	3	2.25	1.125

9	3.222	2.55	1.222
10	3.4	2.8	1.1
Means	1.8319	1.431	0.6479

Pada Tabel 1 ditunjukkan proses perhitungan jumlah obyek *flash disk* yang tidak berhasil terdeteksi menggunakan Vec 1 sebesar 1.8319 obyek dan Vec 2 sebesar 1.431 obyek. Dengan menggunakan Vec gabungan, terdapat penurunan jumlah obyek tak terdeteksi dengan rata-rata 0.6479 obyek yang tidak terdeteksi. Jumlah obyek yang tidak terdeteksi menggunakan *vector* gabungan menurun 64.63% dibanding vector 1 (Vec1) dan 54.72% dibanding vector 2 (Vec2). Akurasi pendeteksian obyek ditunjukkan pada gambar 10.



Gambar 10. Perbandingan Kesalahan Pendeteksian Obyek Vec1, Vec2 dan Vec Gabungan

Uji statistik keberhasilan pendeteksian obyek ditunjukkan pada tabel 2.

Tabel 2. Min, Max, Means Vec Gabungan: N-Pos 40 N-Neg 40 Obyek Terdeteksi

No	Jumlah *Flash disk*	Vec 1: N-Pos 20 N-Neg 40			Vec 2: N-Pos 20 N-Neg 40			Vec Gabungan: N-Pos 40 N-Neg 40		
		MIN	MAX	MEANS	MIN	MAX	MEANS	MIN	MAX	MEANS
1	1	1	1	1	1	1	1	1	1	1
2	2	1	1	1	1	1	1	1	1	1
3	3	1	2	1.3	1	2	1.3	1	2	1.6
4	4	1	2	1.5	1	2	1.5	1	3	2
5	5	0	2	1.8	1	3	1.8	1	5	2.6
6	6	0	2	1.3	1	3	2	1	5	2.8
7	7	0	2	1.428	1	3	2.14	1	5	2.85
8	8	0	2	1.5	1	3	2.25	1	5	3.12
9	9	0	4	1.77	1	4	2.44	1	7	3.55
10	10	0	5	2.1	1	5	2.7	1	10	4.2
Rata-rata Obyek Terdeteksi				14.698			18.13			24.72
				3.2828						2.472
Prosentase				2.472/3.2828x100%= 75.3%						

Tabel 3 merupakan perbandingan antara jumlah obyek terdeteksi dan tidak terdeteksi sebelum dan sesudah proses penggabungan file vektor.

Tabel 3. Hasil Akhir Prosentase Obyek Tidak Terdeteksi

No	Jumlah Obyek Tak Terdeteksi		Jumlah Obyek Terdeteksi	
	Rata-rata Vec 1 dan Vec 2	Vec Gabungan	Rata-rata Vec 1 dan Vec 2	Vec Gabungan
1	1.63145	0.6479	1.6414	2.472
2	1.40525	0.6735	1.8569	2.574
3	1.39185	0.8405	2.1804	2.407
4	1.39185	0.8405	1.4424	2.456
5	1.31535	0.7601	1.8935	2.488
6	1.08825	0.6316	2.109	2.617
7	1.07485	0.366	2.4325	2.882
8	1.3317	0.3517	1.93	2.896
9	1.0896	0.5826	2.1455	2.666
10	1.0762	0.4256	2.469	2.821
11	1.2501	0.5352	1.9965	2.713
Rata-rata	1.27695	0.6050	2.0085	2.635

Pada Tabel 3 ditunjukkan adanya penurunan kesalahan pendeteksian obyek menggunakan vector gabungan dibandingkan dengan rata-rata vector 1 dan vector 2 sebesar 52.62% dan adanya peningkatan pendeteksian obyek sebesar 23.78%

4. Kesimpulan

Berdasarkan hasil dari analisa yang dilakukan pada penelitian ini dapat disimpulkan bahwa:

1. Proses penggabungan *knowledge acquisition* dapat dilakukan dengan menggabungkan file *vector* sebelum proses *training.*
2. Tingkat akurasi pendeteksian obyek menggunakan vector gabungan mengalami peningkatan dengan menurunnya jumlah obyek yang tidak terdeteksi sebesar 52.62% dan meningkatnya jumlah obyek yang terdeteksi sebesar 23.78%

Daftar Pustaka

[1] Jones, M., Viola, P. (2003) *Fast multi-view face detection*. Mitsubishi Electric Research Lab TR-20003-96, 3:14.

[2] Viola, P., & Jones, M. J. (2004). *Robust Real-Time Face Detection. International Journal of Computer Vision*,57(2), 137-154. doi:10.1023/b:visi.0000013087.49260.fb.

[3] Prasetya, D.A, (2012). Deteksi Wajah Metode Viola Jones Pada OpenCV Menggunakan Pemrograman Python. *Simposium Nasional RAPI XI FT UMS.*ISSN : 1412-9612.

[4] Putro, M.D., Adji, T.B., Winduratna, B.(2012). Sistem Deteksi Wajah dengan Menggunakan Metode Viola-Jones. *Seminar Nasional Science, Engineering and Technology*.TIF09-3.

[5] Suharso, Aries,.(2016).*Pengenalan Wajah Menggunakan Viola-Jones dan Eigenface Dengan Variasi Posisi Wajah Bebasis Webcam.* Vol. 1, No. 2, E-ISSN 2503-054X.

[6] Sajati, H. (2018*). The Effect of Peak Signal to Noise Ratio (PSNR) Values on Object Detection Accuracy in Viola Jones Method*. Conference SENATIK STT Adisutjipto Yogyakarta,4, 167-174. doi:10.28989/senatik.v4i0.139

[7] Sianturi, Jonatan. (2018). *Sistem Pendeteksian Manusia untuk Keamanan Rungan Menggunkan Viola-Jone.* Vol. 1, P-ISSN 2549-6247, E-ISSN 2549-6255.

[8] Sun, X., Wu, P., & Hoi, S. C. (2018). *Face detection using deep learning: An improved faster RCNN approach.* Neurocomputing,299, 42-50. doi:10.1016/j.neucom.2018.03.030.

[9] Moghimi, M. M., Nayeri, M., Pourahmadi, M., Moghimi, M. K. (2018). *Moving vehicle detection using AdaBoost and haar-like feature in surveillance videos*, International Journal of Imaging and Robotics, vol. 18, no. 1, pp. 94–106.

[10] Kurniawan, F., Sajati, H., Dinaryanto, O., (2015), *Pendeteksian Kepadatan Lalu-lintas dengan Menggunakan Simpangan Baku Histogram Citra Jalan*. Prosiding Seminar Nasional ReTII ke-10 2015.

[11] Li, S. Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., & Shum, H. (2002). *Statistical Learning of Multi-view Face Detection. Computer Vision* — ECCV 2002 Lecture Notes in Computer Science, 67-81. doi:10.1007/3-540-47979-1_5.

[12] Sajati, H., Astuti, Y., (2013), *Analisis Dan Perancangan Software Untuk Menentukan Warna Kendaraan Gelap Dan Terang*. Jurnal Angkasa Vol. 5 No. 2.

[13] Sajati, H., Astuti, Y., Octaviana, C.H., (2014), *Analisis Pemrosesan Paralel Untuk Kompresi Video Pada Jaringan Komputer Berbasis IPv6*. Jurnal Ilmiah Angkasa Vol. 6, No. 2.

[14] Sajati, H. (2018), *Analisis Kualitas Perbaikan Citra Menggunakan Metode Median Filter Dengan Penyeleksian Nilai Pixel*, Jurnal Ilmiah Angkasa Vol. 10, No. 1, P-ISSN 2085-9503, E-ISSN 2581-1355.

[15] Zein, A. (2018), *Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON*, Sainstech Vol. 28, No. 2, pp. 22-26 ISSN 1410 - 7104