PERANCANGAN SISTEM PARKIR OTOMATIS SUB SISTEM : DETEKSI TANDA MENGGUNAKAN METODA VIOLA-JONES DAN PENGENAL SIMBOL MENGGNAKAN SVM

AUTOMATED PARKING SYSTEM DESIGN SUBSYSTEM : SIGN DETECTION USING VIOLA-JONES METHOD AND SYMBOL RECOGNITION USING SVM

Dimas Gallantino[1], Agus Virgono, Ir.,M.T.[2], Randy Erfa Saputra,S.T.,M.T.[3]

Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

[1] gallantino@student.telkomuniversity.ac.id, [2] avirgono@telkomuniversity.ac.id, [3] ...@telkomuniversity.ac.id

Abstrak

Pada tugas akhir ini telah dirancang sebuah sistem deteksi tanda dan pengenal simbol yang berfungsi sebagai pembantu navigasi sistem kendali robot pengikut garis. Sistem ini berfungsi untuk mendeteksi tanda dan mengenali simbol angka didalam tanda yang menandai posisi setiap slot parkir yang ada. Sistem ini dirancang berdasarkan *real-time object detection framework* yang dirancang oleh Viola dan Jones. Sistem telah dilatih untuk mendeteksi sebuah objek yang telah dirancang sebagai penanda lintasan dan berhasil mencapai tingkat deteksi 96.67% dengan akurasi 95.39% pada pengujiannya di arena simulasi. Untuk mengenal simbol angka digunakan sebuah model klasifikasi yang dilatih menggunakan algoritma SVM dengan *kernel* RBF. Pelatihan dilakukan pada fitur HOG dari data set MNIST beserta beberapa data tambahan berupa angka cetak yang diambil dari 44 *font* berbeda, model klasifikasi yang dilatih berhasil meraih akurasi 93.1624% pada pengujian di arena simulasi.

Kata Kunci : Viola-jones object detection framework, SMV, HOG.

Abstract

In this research has been designed a marker detection and printed digit recognition system that function as a part of navigation system for steering based line follower robot. This system was designed to detect a sign like object that marks position of a parking slot and recognize the printed digit inside the object. The detection system is based on Viola's and Jones's real-time object detection framework. This system has achive detection rate of 96.67% with 95.39% accurcy on a test in simulation arena. To recognize the digits inside the object, a classification model is trained using SVM algorithm with RBF kernel. This classification model is trained on HOG features of MNIST data set and some additional data from 44 diffrent font of printed digit. This classification sistem achive 99.37% accuracy on a test in a simulation arena.

Key Words : Viola-jones object detection framework, SMV, HOG.

1. Pendahuluan

Memarkikan kendaraan merupakan kegiatan yang memakan waktu [1] dan harus dilakukan oleh setiap pengendara kendaraan beroda empat. Untuk mengurangi beban pengendara dalam hal tersebut sistem management *smart parking* [2] [3] dapat diterapkan. Dengan sistem seperti ini keterlibatan manusia masih diperlukan dan memungkinkan adanya kecelakaan yang memakan korban jiwa baik dari pejalan kaki di area parkir atau pengendara yang sedang memarkirkan kendaraannya [4].

Salah satu cara mengatasinya dapat menggunakan sebuah sistem parkir otomatis, dimana sistem ini dapat memanfaatkan informasi mengenai nomor tempat parkir yang tersedia, seperti informasi yang dihasilkan oleh sistem rancangan Khanna [3], kemudian menggunakannya untuk memakirkan kendaran secara otomatis. Sistem seperti ini akan membutuhkan sebuah sub-sistem navigasi yang dapat memberikan informasi yang diperlukan untuk memarkirkan kendaraan. Salah satu informasi yang dapat digunakan adalah tanda-tanda jalan yang digunakan untuk menandai tempat parkir. Dengan mendeteksi dan mengenali arti tanda-tanda ini sistem parkir otomatis dapat mengetahui posisi tempat parkir yang dituju.

Perancangan sebuah sistem parkir otomatis untuk mobil seperti ini dapat dilakukan terlebih dahulu pada sebuah prototipe menggunakan robot mobil mini [5]. Pada tugas akhir ini diajukan sebuah rancangan sub-sistem pendeteksi tanda dan pengenal simbol pada citra digital yang didapat dari kamera yang dipasang di badan robot protipe mobil mini. Rancangan sistem ini dapat dibagi menjadi dua bagian utama, bagian pertama adalah detektor yang befungsi untuk mendeteksi tanda pada jalur, bagian kedua merupakan sistem klasifikasi yang berfungsi untuk mengenal simbol yang ada di dalam tanda yang telah terdeteksi.

2. Dasar Teori

2.1 *Viola and Jones Rapid Object Detection Framwork*

Metoda deteksi ini pertamakali dikembangkan oleh Viola dan Jones pada tahun 2001 [6]. Metoda ini bekerja dengan memindai citra atau potongan citra dengan sebuah ukuran jendela, selama pemindaian sub-area yang berada di dalam jendela ini diklasifikasi sebagai objek atau latar belakang menggunakan sebuah sistem klasifikasi bertahap. Pada setiap tahap di sistem ini terdapat fungsi klasifikasi yang dapat mengklasifikasikan potongan citra sebagai objek atau tidak. Sub-area yang diklasifikasikan sebagai objek pada setiap tahap di sistem klasifikasi bertahap akan dikategorikan sebagai objek dan jika sub-area diklasifikasikan sebagai non-objek disebuah stage sub-area tersebut akan diabaikan. Dengan cara klasifikasi seperti ini metoda deteksi ini dapat mendeteksi objek dengan cepat karena perhitungan pada tahap-tahap akhir dilakukan pada sub-area yang lebih menjanjikan. Model klasifikasi pada setiap tahap didalam sistem klasifikasi bertahap ini adalah sebuah *strong classifier* yang terdiri dari kumpulan beberapa sistem klasifikasi sederhana yang bergantung pada satu buah fitur HAAR-*like*.

2.2 *Histogram of Oriented Gradient*

Histogram of Oriented Gradient (HOG) merupakan sebuah algorima ekstrasi fitur padat yang pertamakali dikembangkan oleh Dalal [7]. Fitur HOG diambil dari sebuah citra atau sebuah Region Of Interest (ROI) di dalam citra secara keseluruhan. Fitur yang diekstrasi algorima ini merepresentasikan arah gradien intensitas pixel atau gradien warna dan juga magnitude gradiennya dalam bentuk histogram.

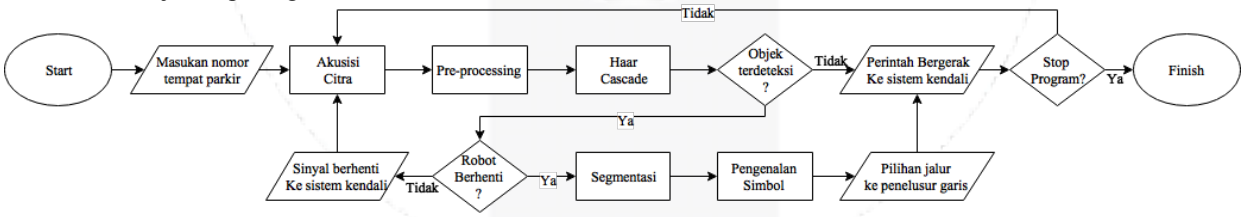
2.3 *Support Vector Machine*

Support Vector Machine (SVM) merupakan sebuah algoritma pembelajaran klasifikasi biner yang pertama kali diusulkan oleh Vapnik [8]. Algortima ini menyelesaikan sebuah masalah kelasifikasi biner dengan mencari sebuah *hyperplane* pemisah yang memiliki margin terluas, dimana margin merupakan jarak antara hyperplane dan data poin atau vektor terluar dari kedua kelas. Masalah ini diformulasikan sebagai masalah optimasi besar margin *hyperplane* dengan *constraint* hyperplane tersebut harus bisa memisahkan dua kelas data dengan benar, atau dalam kasus *soft-margin* SVM C-SVM dimana kesalahan kelasifikasi pada data latih di perbolehkan formulasi masalah pembelajaran SVM menjadi optimasi luas margin dan juga optimasi kesalahan kelasifikasi datalatih. Pada implementasinya pelatihan sebuah model C-SVM dilakukan dengan menyelesaikan *lagrange dual* dari permasalahan optimasi primal, dengan demikian menyelesaikan permasalahan optimasi menjadi lebih sederhana.

3. Perancangan Sistem

Sistem yang dirancang secara kesulurahan merupakan sebuah bagian dari sistem nafigasi robot mobil mini yang berfungsi untuk mendeteksi dan mengenal tanda pada arena simulasi yang telah dirancang khusus. Di sepanjang jalur lintasan terdapat tanda yang akan dideteksi dan dikenali oleh sistem pendeteksi tanda dan pengenal simbol. Keluaran dari sistem ini digunakan robot untuk menentukan jalur yang harus dipilih didalam arena.

Sistem yang dirancang akan diintegrasi dengan sistem penelususr garis dan juga sistem kendali robot. Sistem ini berkomunikasi dengan sistem penelusur garis untuk memberitahu jalur mana yang harus dipilih ketika sistem tersebut mendeteksi ada dua jalur atau lebih yang dapat diikuti. Sistem ini juga berkomunikasi dengan sistem kendali robot untuk memberitahu kapan robot harus berehenti dan kapan robot bisa bergerak kembali. Alur kerja sistem ini ditunjukan pada gambar 3.1.

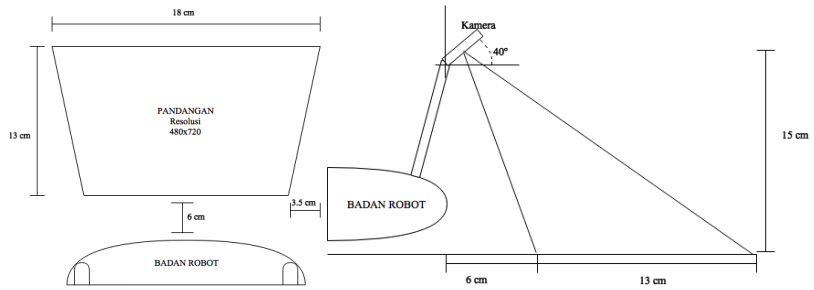


Gambar 3.1 Diagram alir sistem deteksi dan pengenal simbol.

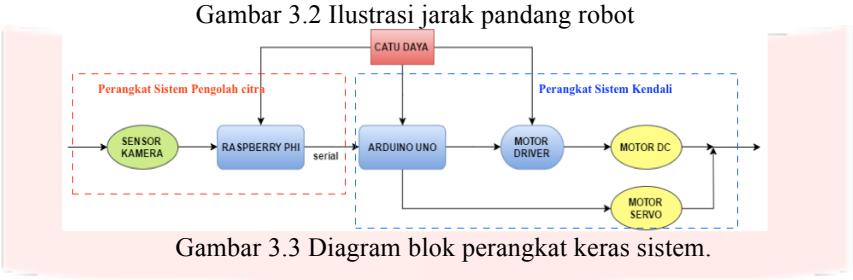
3.1 Perancangan Perangkat Keras

Pada tugas akhir ini digunakan Raspberry Pi 3B+ untuk melakukan proses pengolahan citra dan proses lainnya yang dilakukan sistem. Modul Raspberry Pi *camera board* dipasangkan di badan robot sebagai perangkat untuk akusisi citra, modul ini dipasang pada bagian depan robot mobil mini dengan derajat kemiringan ±40° relatif dengan dasar robot dan tinggi ±15cm dari dasar robot, akusisi citra dilakukan dengan resolusi 480 pixel x 720 pixel. Pada gambar 3.2 ditunjukan ilustrasi jarak pandang robot.

Sistem pendeteksi tanda pada Raspberry Pi berkomunikasi dengan sistem kendali robot melalui pin GPIO untuk mengirimkan sinyal berhenti kepada sistem kendali berupa tegangan listrik 3.3v. Robot perlu dihentikan lajunya saat sebuah tanda terdeksi untuk mengurangi kemungkinan adanya distorsi pada citra yang akan diambil selatjutnya yang disebabkan pergerakan kamera. Rancangan perangkat keras ditunjukan pada gambar 3.3.



Gambar 3.2 Ilustrasi jarak pandang robot



Gambar 3.3 Diagram blok perangkat keras sistem.

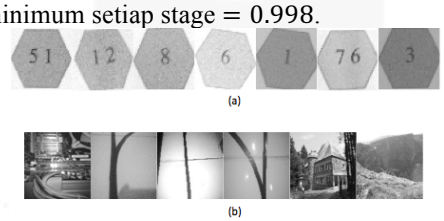
3.2 Sistem Deteksi

Sistem deteksi tanda yang dirancang pada penelitian ini merupakan sistem pendeteksi objek *real-time* berbasis pada *framework* yang dirancang oleh Viola dan Jones [6], dalam penelitian ini *framework* tersebut akan direferensi sebagai HAAR *cascade*. Sebelum sistem deteksi dijalankan, format citra RGB (Red Green Value image) yang didapat dari sensor kamera diubah ke *grayscale*. Kemudian pemindaian dengan HAAR *cascade* dilakukan dengan ukuran jendela yang diperbesar 10% pada setiap iterasinya.

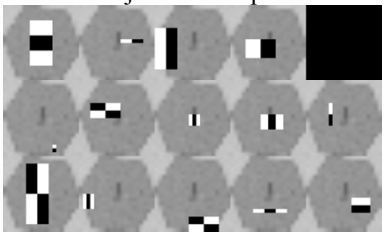
Objek tanda yang digunakan berukuran 3.5cm x 3.5cm, saat robot bergerak menulusuri garis objek didalam frame ukurannya akan bervariasi kurang lebih dari 150x150 pixel sampai 230x230 pixel di dalam frame. Untuk mempercepat proses pemindaian, ukuran jendela pemindaian minimum dan maksimum dibatasi dengan ukuran 100pixel x 100pixel sampai 300x300 pixel sehingga pemindaian tidak perlu dilakukan menggunakan ukuran jendela yang terlalu kecil atau terlalu besar.

Untuk melatih model HAAR *cascade* digunakan algoritma yang dikembangkan oleh Viola dan Jones [6]. algorima ini memerlukan inisialisasi untuk melatih HAAR *cascade*. Inisialisasi yang diberikan pada algorima untuk melatih sistem deteksi pada penelitian ini ditunjukan pada poin-poin berikut.

- Himpunan 2000 citra positif berukuran 20x20 pixel yang berisikan objek, yaitu tanda yang ingin di deteksi, Sample ditunjukan pada gambar 3.4 (a).
- Himpunan 4000 citra negatif berukuran 500x500 pixel berisikan citra latar belakang dimana objek yang ingin di deteksi tidak dapat di temukan di antara kumpulan citra-citra ini, sample ditunjukan pada gambar 3.4 (b).
- Rasio *false alarm* maximum setiap stage $= 0.35$.
- Rasio *false alarm* maximum yang ingin dicapai $= 5\times10^{-7}$.
- Rasio *true positive* minimum setiap stage $= 0.998$.



Gambar 3.4 Sampel citra pelatihan yang digunakan. (a) sampel citra positif yang berisikan objek yang inging dideteksi, citra ini ditambahkan *background* dan diperkecil resolusinya menjadi 20x20 *pixel* sebelum digunakan. (b) sampel citra negatif dimana objek tidak dapat ditemukan di citra-citra ini..



Gambar 3.5 Hasil pelatihan Cascade Classifier. Visualisasi fitur-fitur yang dipilih adaboost pada stage 1 dan 11.

Dalam praktiknya untuk melatih *cascade classfier* dilakukan menggunakan aplikasi berbasis teks dari OpenCV bernama opencv_traincascade dan memberikan masukan untuk setiap parameter. Model cascade classifier yang dihasilkan dari proses pelatihan menggunakan 11 stage dan 121 fitur. Fitur-fitur pada stage 1 dan 11 ditunjukan pada gambar 11.

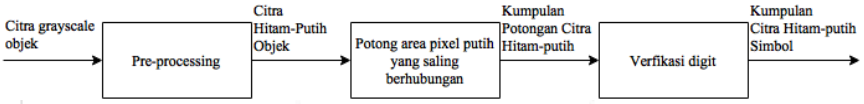
3.3 Segmentasi Simbol

Setiap kali objek terdeteksi pada sebuah frame t maka sistem deteksi objek akan mengirimkan sinyal ke sistem kendali untuk memberhentikan laju robot dan sistem akan diberhentikan sejenak untuk menunggu robot berhenti. Jika pada frame t+1 objek terdeteksi kembali dan robot sedang berhenti maka pemrosesan citra akan dilakukan pada *Region of Interest* (ROI) di dalam citra yang didapat dari HAAR *cascade*, jika tidak sistem kendali akan diberikan sinyal untuk melaju kembali. Pemrosesan ini bertujuan untuk mendapatkan citra hitam putih dari objek tanda sehingga simbol angka mudah disegmentasi. Proses pengolahan citra ditunjukan pada poin-poin berikut:

- Median Blur dengan ukuran kernel 7 *pixel*.
- Operasi titik, pengalian dengan faktor pengali 3.
- MSER Detection.

Citra hitam putih yang didapatkan dari pemrosesan citra dipotong berdasarkan area *pixel* putih yang berdampingan. Setiap potongan citra hitam putih yang didapat dari proses segmentasi akan dihitung beberapa nilai propertinya untuk memastikan citra yang didapat merupakan citra digit, nilai-nilai ini adalah kepadatan, *extent* dan rasio aspek. Seluruh proses ini digambarkan pada diagramblok didalam gambar 3.6. Potongan citra akan dianggap sebagai citra angka jika memenuhi peraturan–peraturan dibawah ini.

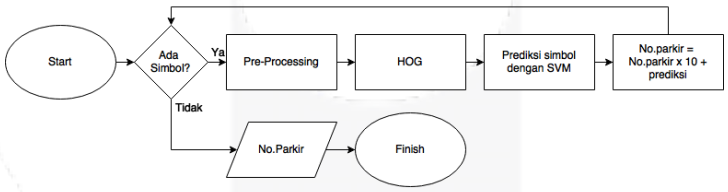
- 0.3 < Rasio Aspek < 0.875.
- 0.5 < Kepadatan < 0.95
- 0.5 < *Extent* < 0.85



Gambar 3.6 Diagram blok proses segmentasi

3.4 Pengenalan Simbol Digit

Pada sitem ini sebuah model klasifikasi dilatih menggunakan algoritma SVM untuk mengenal pola fitur pada citra simbol angka yang didapat dari proses segmentasi. model Klasifikasi dilatih menggunakan fitur HOG dari data pelatihan. Untuk memprediksi citra simbol angka yang didapat dari proses segmentasi, citra ini pertama ditambahkan lapisan pixel hitam di sekelilingnya sampai lebar dan panjang citra menjadi sama besarnya, kemudian citra diubah ukurannya menjadi 28x28 pixel. Dari citra ini fitur HOG diambil dan diberikan ke sistem klasifikasi SVM untuk memprediksi citra simbol. Dan proses ini dilakukan untuk setiap citra yang didapat dari proses segmentasi untuk mendapatkan nomor slot parkir dari objek yang terdeteksi. Alur kerja sistem ini ditunjukan pada diagram gambar 3.7.



Gambar 3.7. Diagram alur prediksi simbol pada tanda

3.4.1 Pelatihan SVM

Pada tugas akhir ini LIBSVM [9] yang diimplementasikan di OpenCV digunakan untuk melatih dan menjalankan model klasifikasi C-SVM. Untuk melatih sebuah model klasifikasi SVM membutuhkan beberapa nilai parameter yaitu nilai C, kernel yang digunakan dan nilai parameter untuk setiap kernel yang digunakan. Optimasi untuk parameter-parameter ini dilakukan dengan menginisialisasi beberapa argumen untuk setiap parameter. Kemudian untuk setiap kombinasi argumen parameter model klasifikasi dilatih dan dievaluasi, parameter optimal kemudian dilpilih berdasarkan analisis hasil evaluasi ini. Poin-poin di bawah ini menunjukan parameter yang dioptimasi dan argumen yang dipilih berdasarkan penelitian sebelumnya [8] [10] ditambah beberapa argumen lain untuk perbandingan.

- C ∈ {0.1, 1, 10, 100, 300, 500 }.
- Kernel ∈ {Radial Basis Function(RBF)/gaussian, Polynomial, Linier}.
- Gamma(γ) ∈ {0.00001, 0.0025, 0.05, 0.2, 0.5}.
- Degre(σ) ∈ {0.6, 1, 2, 2.5, 3}.
- Coef0 ∈ {0.1, 0.5, 1, 2.5, 5}.

3.4.2 Data Latih SVM

Data pelatihan MNIST merupakan kumpulan 60000 citra grayscale digit angka tulisan tangan dengan resolusi 28x28, Model SVM dilatih menggunakan data MNIST dan juga data angka cetak yang dibentuk dari 44 font berbeda dan digandakan hingga berjumlah 10000 dengan resolusi yang sama dengan data MNIST. Dari data ini fitur HOG diambil dari setiap sampel data pelatihan. Kumpulan vektor fitur inilah yang digunakan untuk pelatihan.

Fitur HOG diambil dengan sel berukran 7x7 pixel. *Histogram* untuk setiap sel dibentuk dalam format orientasi *9bin*, sehingga untuk setiap sel, setiap histogram yang dibentuk akan memimiliki sembilan nilai yang merepresentasikan orientasi dari 0 sampai 360 derajat jika menggunakan *signed gradient* atau 0 sampai 180 derajat jika menggunakan *unsigned gradient*. Bagian 4.2 membahas efek penggunaan *signed gradient* dan *unsigned gradien* pada pelatihan model.

Normalisasi *histogram* dilakukan menggunakan ukuran blok 14x14 pixel. blok ini digeser keseluruh gambar seperti memindai dengan langkah pergeseran 7x7 pixel. Karna di setiap pergeseran blok terdapat 4 *histogram* yang dinormalisasi setiap pergeseran akan membentuk vektor dengan 36 dimensi. Pada setiap citra blok digeser sebanyak 9 kali dengan demikian dari setiap citra pada dataset akan menghasilkan vektor fitur dengan 324 dimensi.

4. Pengujian

Disini setiap aspek dari sistem seperti perangkat keras, sistem deteksi dan sistem pengenal simbol digit diuji dan analisa performansinya untuk mengetahui parameter terbaik untuk setiap sistem, kemudia sistem dengan parameter tersebut diuji dan dianalisa secara kesulurahan.

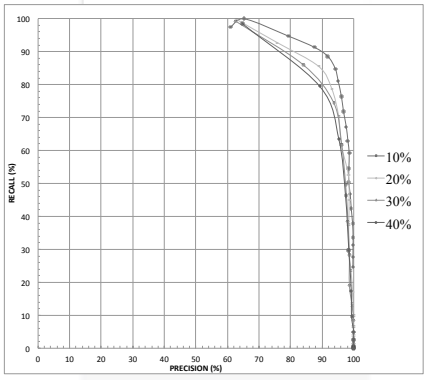
4.1 Pengujian Sistem Deteksi.

Pengujian ini bertujuan untuk mengetahui pengaturan parameter terbaik untuk sistem deteksi, kemudian pengaturan tersebut digunakan untuk menguji sistem secara keseluruhan. Parameter-parameter ini adalah nilai pengali untuk memperbesar jendela pemindaian dan nilai keyakinan minimum deteksi. Pengujian dilakukan menggunakan sebuah data yang telah dirancang khusus. Data pengujian ini merupakan kumpulan 1500 citra latar belakang berukuran 500x500 pixel dengan satu buah objek tanda yang posisi, ukuran, intensitas warna dan orientasinya berbeda beda.

Setiap objek didalam citra telah diketahui posisi dan ukurannya, informasi ini digunakan untuk dibandingkan dengan keluaran sistem deteksi sehingga jumlah hasil deteksi yang salah dan juga yang benar dapat diketahui. Dari pengujian ini dapat dihitung nilai *precession* dan *recall* untuk mengukur performansi sistem.

$$Precession = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$



Gambar 4.1 Grafik Precession-Recall sistem deteksi tanda.

Setiap kurva di dalam grafik pada gambar 4.1 didapat dari parameter faktor pengali ukuran jendela yang berbeda dan setiap poin di atas kurva didapat dari nilai keyakinan minimum yang berbeda. Dapat dilihat pada grafik ini bahwa semakin kecil faktor pengali ukuran jendela maka semakin baik kerja sistem deteksi, hal ini ditunjukan oleh area dibawah kurfa yang semakin besar. Dengan semakin kecilnya faktor pengali ukuran jendela maka semakin banyak ukuran jendela pemindai berbeda yang digunakan untuk memindai citra, dengan demikian semakin besar pula kemungkinan sistem deteksi menemukan objek.

Dari grafik pada gambar 4.2 hasil pengujian dengan akurasi yang lebih tinggi adalah hasil pengujian dengan nilai keyakinan minimum yang lebih besar. Hal ini terjadi karna sistem deteksi lebih yakin dengan hasil deteksinya, tetapi hal ini juga menurunkan tingkat deteksi karna adanya objek yang terdeteksi dengan benar tetapi nilai keyakinan hasil deteksi kurang dari nilai minimum. Sebaliknya semakin kecil nilai keyakinan minimum yang gunakan semakin tinggi pula tingkat deteksi sistem. Tetapi hal ini juga menyebabkan semakin banyak deteksi yang salah dan menurunkan akurasi karena sistem tidak perlu terlalu yakin dengan hasil deteksinya.

Sebuah model sistem deteksi yang ideal akan berada pada titik (100, 100) di dalam grafik ini. Dengan demikian untuk memilih parameter terbaik untuk model deteksi ini dapat dilakukan dengan memilih parameter yang menghasilkan nilai *precession-recal* paling dekat dengan titik (1,1). Tabel 4.1 menunjukan beberapa hasil pengujian dan parameter yang digunakan, hasil pengujian yang mendekati sistem deteksi ideal ditunjukan pada data yang dicetak tebal.

Tabel 4.1 Beberapa data poin pada grafik precession-recall mendekati poin (100, 100).

n	Faktor Pengali Ukuran Jendela (a)			
	10%		20%	
	Prec %	Rec %	Prec %	Rec %
3	87.5293	91.3256	93.0319	0.787822
4	**91.8328**	**88.5307**	95.2257	0.704108
5	94.1423	84.8523	96.0646	0.618583

4.2 Pengujian Sistem Kelasifikasi

Optimasi parameter dilakukan dengan memilih beberapa argumen yang dapat digunakan untuk setiap parameter. Kemudian untuk setiap kombinasi argumen yang dapat digunakan sebuah model C-SVM dilatih dan dievaluasi. Parameter optimal dipilih berdasakan hasil evaluasi ini. Hasil evaluasi bertujuan untuk mendapatkan tingkat kesalahan dan juga tingkat akurasi klasifikasi yang dihitung menggunakan rumus (3) dan (4).

$$Tingkat\ akurasi = \frac{jumlah\ prediksi\ yang\ benar}{jumlah\ prediksi\ yang\ benar + jumlah\ prediksi\ yang\ salah} \times 100 \quad (3)$$

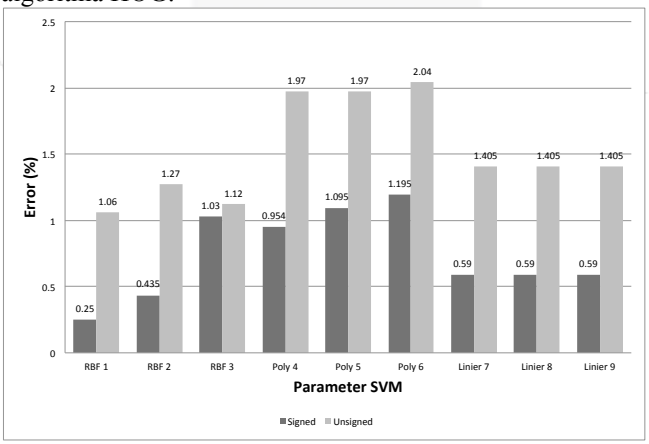
$$Tingkat\ kesalahan = 100 - Tingkat\ akurasi \quad (4)$$

Pada pengujian untuk pencarian parameter optimal setiap model C-SVM dilatih menggunakan dataset pelatihan MNIST dan juga digit cetak, fitur yang digunakan untuk pelatihan diambil menggunakan algoritma HOG dengan *signed gradien*.

Tabel 4.2 Beberapa pengaturan parameter SVM yang dipilih

No	*Kernel*	γ	σ	Coef0	C	Err %
1	RBF	0.5	-	-	10	0.25
2		0.0025	-	-	500	0.435
3		0.2	-	-	1	1.03
4	Poly	0.05	2	5	10	0.945
5		0.05	2	5	100	1.095
6		0.2	2	5	300	1.195
7	Linier	-	-	-	100	0.59
8		-	-	-	300	0.59
9		-	-	-	500	0.59

Pada tabel 4.3 telah dipilih beberapa parameter yang meminimalisasi kesalahan deteksi pada data pengujian. Dengan menggunakan parameter-parameter ini model C-SVM dilatih kembali untuk mengetahui efek dari encoding gradient yang berbeda pada algoritma HOG.



Gambar 4.2. Perbandingan tingkat kesalahan antara pemrosesan data latih

Pada gambar 4.3 menunjukan tingkat kesalahan untuk setiap model C-SVM yang dilatih dengan parameter pada tabel 4.2 menggunakan cara ekstrasi fitur yang berbeda. Disini dapat dilihat pelatihan menggunakan fitur yang dimabil algoritma HOG dengan tipe encoding gradien *unsigned* perfomanya menurun dibandingkan dengan *signed*. Pada penelitian sebelumnya menggunakan Pyramid HOG(PHOG) [10] efek yang sama juga muncul, pada penelitian tersebut dijelaskan signed gradien lebih baik mendeskripsikan citra seperti pada dataset MNIST karena tipe encoding ini dapat membedakan gradien dari terang ke gelap dan gelap ke terang. Dari sini disimpulkan bahwa data set yang optimal adalah data yang fiturnya di ambil menggunakan HOG signed gradient.

Dari setiap pengujian, model C-SVM dengan kernel RBF menggunakan gamma = 0.5 dan C=10 adalah model C-SVM yang terbaik karna model ini memiliki tingkat kesalahan klasifikasi terkecil. Model inilah yang diimplementasikan pada sistem klasifikasi. Tabel 4.3 menunjukan perbandingan model klasifikasi pada penelitian ini dengan penelitian sebelumnya yang dilatih dan diuji dengan data MNIST saja.

Tabel 4.3 Perbandingan model klasifikasi pada penelitian sebelumnya

Model kelasifikasi	Fitur	Dimensi Fitur	Err %
SVM, RBF kernel (gamma = 0.5) (c=10)	**HOG**	**324**	**0.63**
SVM, Polynomial kernel [10] (orde = 5)(c=10)	Pyiramid HOG(PHOG)	2869	0.56
SVM, Linear [11]	9x9 sel HOG	81	2.75

Dibandingkan dengan penelitian sebelumnya yang juga menggunakan fitur HOG sederhana [11], terlihat perkembangan yang signifikan dapat di capai hanya dengan mengubah ukuran sel. Dengan metoda ekstrasi fitur PHOG [10] model klasifikasi dapat mecapai tingkat akurasi 99.44%. Walaupun demikian dimensi vektor fitur yang dihasilkan sangatlah tinggi sehingga akan memperlambat waktu pelatihan dan juga prediksinya. Pada tugas akhir ini di pilih ekstrasi fitur yang lebih sederhana karna implementasinya pada sebuah sistem *real-time*.

4.3 Pengujian Sistem Secara Keseluruhan.

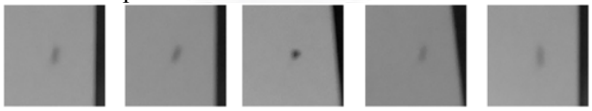
Pengujian ini dilakukan pada lintasan lurus sepanjang 150 cm. Pada lintasan terdapat 5 objek dengan jarak ±30cm. Sistem harus dapat mendeteksi tiap tanda 2 kali, saat bergerak ketika objek secara penuh memasuki frame dan saat robot berhenti. Analisi akan dilakukan pada citra objek ketika sitem mendteksi objek dan citra simbol angka Disini nilai Pulse Width Modulation (PWM) pada arduino digunakan untuk mengubah kecepatan robot. Pada setiap kecepatan yang berbeda percobaan dilakukan beberapa kali untuk mendapatkan data yang cukup untuk analisa. Dalam percobaan nilai akurasi deteksi dihitung dengan persamaan (1). sedangkan tingkat deteksi dihitung dengan persamaan (3).

$$Recall = \frac{True\ Positive}{Detection\ Target} \quad (5)$$

Tabel 4.4 Hasil pengujian dengan nilai PWM berbeda.

PWM	Iterasi	Detection Target	True Positif	False Positive	Precession	Recall
80	16	160	156	14	0.9123	0.975
90	15	150	145	7	0.9539	0.9667
100	15	150	89	18	0.83177	0.5933

Kegagalan deteksi dalam percobaan ini sebagian besar terjadi saat robot sedang berhenti, deteksi saat berhenti gagal dikarnakan robot memakan waktu terlalu lama untuk berhenti sehingga objek tidak dapat tertangkap oleh kamera secara keseluruhan atau sama sekali walaupun objek masih berada di depan badan robot. Pada pengujian dengan nilai PWM = 100, keberhasilan deteksi hanya mencapai 59.3% dan 82% dari seluruh objek yang terdeteksi di deteksi saat robot bergerak. Dalam percobaan ini hanya 2 objek yang di lewati secara keseluruhan oleh robot, hal ini terjadi ketika posisi start robot tidak lurus dengan jalur dan robot sedikit berosilasi di awal untuk menstabilkan laju dan membuat objek tidak terlihat secara penuh didalm frame saat robot berada di dekat objek.



Gambar 4.3 Sample dari 54% kesalahan deteksi

54% kesalahan deteksi dalam pengujian ini terjadi ketika sistem deteksi mengenal noise pada lintasan sebagai objek, noise ini memiliki fitur yang terlihat mirip oleh sistem deteksi dengan fitur paling dominan pada objek yang ingin di deteksi (gambar 4.4), hal ini juga terjadi pada pengujian sebelumnya menggunakan daset pengujian pada bagian 4.2. Pada pengujian dengan PWM =100 kecepatan yang lebih tinggi menyebabkan noise tersebut semakin terlihat seperti simbol digit di dalam tanda sehingga akurasi semakin menurun. pada pengujian dengan nilai PWM = 90 noise pada lintasan dihilangkan sehingga akurasi meningkat.

Dari tabel 4.8 dapat dilihat sebagian besar kesalahan disebabkan oleh proses segmentasi sehingga membuat citra sulit untuk di kenali, kegagalan segmentasi ini biasanya terjadi pada iluminasi yang tidak merata, Dari 234 segmentasi simbol yang dilakukan sistem selama proses pengujian terdapat 16 simbol yang salah di klasifikasikan oleh sistem klasifikasi, sehingga presentase rasio kesalahan prediksi dengan total prediksi adalah:

$$Err\ Rate = \frac{kesalahan\ prediksi}{total\ prediksi} = \frac{16}{234}\ 100\% = 6.8376\%$$

Tabel 4.5 Kesalahan prediksi sistem klasifikasi.

Prediksi Sistem	Citra yang diprediksi
1	
2	
4	
5	
7	

5. Daftar Pustaka

[6] Paul Viola and Michael J Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, pp. 137–154, September 2001.

[8] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273--297, Sep. 1995.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* , vol. 1, pp. 886-893, Juni 2005.

[1] Donald Shoup, "Cruishing for Parking," *Transport Policy*, vol. 13, pp. 479–486, 2006.

[3] Abhirup Khanna and Rishi Anand, "IoT based Smart Parking System," in *2016 International Conference on Internet of Things and Applications (IOTA)*, Pune, September 2016, pp. 266-270.

[4] Paul Green, "Parking Crashes and Parking Assistance System Design: Evidence from Crash Databases, the Literature, and Insurance Agent Interviews," *SAE Technical Paper*, no. 2006-01-1685, 2006.

[12] Robert E Schapire Yoav Freund, "A Decesion-Theoritic Generalization of On-Line Learning and an Aplication to Boosting," *Journal of Computer and System Sciences*, no. 66, pp. 119 - 139, 1997.

[9] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, April 2011.

[11] Reza Ebrahimzadeh and Mahdi Jampour, "Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM," *International Journal of Computer Applications*, vol. 104, no. 9, oktober 2014.

[2] Hilal Al-Kharusi and Ibrahim Al-Bahadly, "Intelligent Parking Management System Based on Image Processing," *World Journal of Engineering and Technology*, vol. 2, pp. 55-67, Mei 2014.

[5] Yusmansyah, Erwin Susanto, and Angga Rusdinar, "Perancangan dan Implementasi Sistem Kontrol Parkir Mobil Listrik Otomatis Menggunakan Metoda Ackerman Steering," in *e-Proceeding of Engineering*, vol. 2, Bandung, agustus 2015, pp. 2158 -2165.

[10] Subhransu Maji and Jitendra Malik, "Fast and Accurate Digit Classification," Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report 2009.