# IMPLEMENTASI METODE KLASIFIKASI SUPPORT VECTOR MACHINE DAN BACKPROPAGATION UNTUK DETEKSI MOBIL MENGGUNAKAN HISTOGRAM OF ORIENTED GRADIENTS

Muhammad F R kaisupy[1], Galih Hermawan[2]
[1,2] Universitas Komputer Indonesia
Jl. Dipatiukur No 112-116 Bandung. 40132
E-mail : ramdhany.kaisupy@gmail.com, galih.hermawan@email.unikom.ac.id,

## ABSTRAK

Sistem deteksi merupakan suatu teknologi yang sangat membantu dalam berbagai bidang. Dalam bidang lalulintas, sistem deteksi dipakai untuk pemantauan traffic lalulintas. Saat ini pengawasan lalu-lintas telah memanfaatkan kamera CCTV yaitu RTMC (*Road Traffic Monitoring Center*). Selain itu penelitian tentang ITS (*Intelligent Transportation System*) juga memanfaatkan pengawasan melalui kamera.

*HOG* digunakan dalam penelitian ini dikarenakan merupakan Feature Descriptor yang mengambil tepi atau struktur gradient yang terkarakteristik dari bentuk lokal atau arah tepi dan dengan distribusi intensitas gradient lokal yang baik. Hasil ekstraksi feature akan ditraining menggunakan *SVM* untuk menghasilkan suatu model sebagai acuan deteksi kemudian. Hasil deteksi akan diklasifikasi kembali oleh *neural network backpropagation*.

Pengujian dilakukan menggunakan 400 data positif dan 1400 data negatif menunjukan akurasi sebesar 51.8% dengan menggunakan 1 neuron pada hidden layer..

**Kata kunci :** *Backpropagation*, Deteksi mobil, *Support Vector Machine* , *Histogram of oriented Gradient*

## 1. PENDAHULUAN

Beberapa riset tentang deteksi kendaraan pada video pernah dilakukan salah satunya adalah dengan menggunakan metode backgrund substraction yakni membandingkan citra background tanpa objek dengan video, masalah yang didapat pada penelitian ini adalah adanya bayangan yang ditimbulkan akibat cahaya matahari yang mengenai kendaraan akan terdeteksi sebagai objek.

Terdapat beberapa metode yang biasa digunakan untuk deteksi pada pengolahan citra, salah satunya adalah HOG(histogram of oriented gradient). HOG digunakan dalam penelitian ini dikarenakan menurut penelitian yang dilakukan oleh ari kurniawan [1] HOG merupakan Feature Descriptor yang mengambil tepi atau struktur gradient yang terkarakteristik dari bentuk lokal atau arah tepi dan dengan distribusi intensitas gradient lokal yang baik selain itu penelitian yang dilakukan oleh Vanda Singh [2] mendapat hasil feature eksraksi menggunakan metode histogram of oriented gradient lebih baik daripada menggunakan LBP.

Pada penelitian oleh Permata Cahyo, I Ketut eddy, dan Muhtadin [3] dengan menggunakan klasifikasi SVM , didapatkan hasil akurasi tes model sebesar 99% sedangkan hasil deteksi mobil mendapat akurasi 76,17%, karena itu sistem deteksi masih membutuhkan metode baru untuk meningkatkan performa akurasi, beberapa penelitian juga dilakukan untuk meningkatkan performa deteksi histogram of oriented gradient yaitu dengan cara mengkombinasikan beberapa metode klasifikasi seperti penelitian yang dilakukan oleh K.P Sanak Kumar dan Dr. R. Bhavani [4] dimana didapatkan hasil HOG dengan SVM mendapat akurasi 70.85%, sedangkan HOG dengan kNN mendapat akurasi 68.21%, dan HOG dengan SVMkNN mendapat akurasi 72.48%.

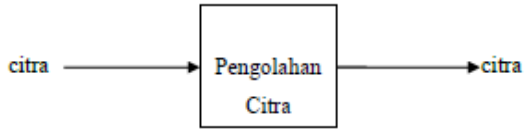
Pada penelitian yang dilakukan oleh Indu Dokare dan Naveeta Kant [5] yaitu membandingkan performa dari metode klasifikasi SVM, KNN dan BPNN didapatkan hasil klasifikasi menggunakan BPNN mendapat akurasi paling tinggi , dan metode klasifikasi kNN mendapat akurasi paling rendah. Kombinasi SVM dan BPNN untuk klasifikasi text pernah dilakukan pada penelitian Zhang Wen, Xijin Tang dan Taketoshi Yoshida [6], dimana didapatkan hasil klasifikasi menggunakan SVM dan BPNN lebih baik daripada menggunakan SVM atau BPNN.

## 2. ISI PENELITIAN

### 2.1 Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra dengan menggunakan komputer agar kualitasnya lebih baik. Tujuannya agar citra dapat dengan mudah diinterpretasi oleh manusia atau komputer. Pada gambar 1 digambarkan bahwa teknik pengolahan citra memanipulasi citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra. Namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan.

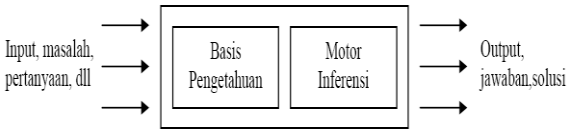
Pada umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila [8]:




Gambar 1. Pengolahan Citra

a. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas
penampakan atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra,
b. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
c. Sebagian citra perlu digabung dengan citra yang lain.

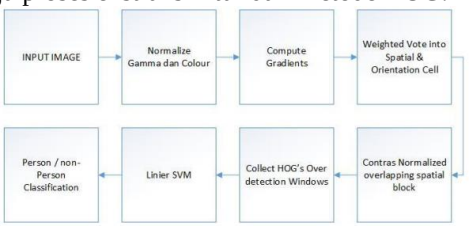
Sub bab ditulis dengan font times new roman ukuran 10. Gmabar dan tabel dapat dimasukkan dibagin ini dengan format seperti pada Gambar 1.




Gambar 1. Contoh penulisan Gambar

## 2.2 Histogram of Oriented Gradients

Histograms of Oriented Gradients (HOG) adalah pendeskripsi ciri (feature descriptor) yang digunakan dalam computer vision dan image processing untuk pendeteksian objek. Metode HOG diperkenalkan pertama kali oleh Dalal dan Triggs pada tahun 2005 . Metode HOG dalam penelitian ini akan digunakan untuk menguji apakah dalam citra uji terdapat manusia atau tidak. Gambar 2 menunjukkan langkah kerja proses ekstraksi fitur dari metode HOG.



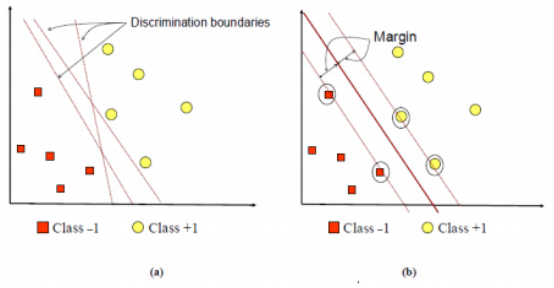

Gambar 2 Proses Ekstraksi fitur HOG

## 2.3 Support vector machine

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition. SVM adalah metode learning machine yang bekerja atas prinsip Structural Risk Minimization (SRM) dengan tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space [7].

SVM dalam penelitian ini termasuk dalam rangkaian metode HOG yang merupakan alat untuk menetapkan apakah dalam detection window terdapat sosok manusia atau tidak. SVM classifier berfungsi untuk memisahkan class manusia dan nonmanusia.

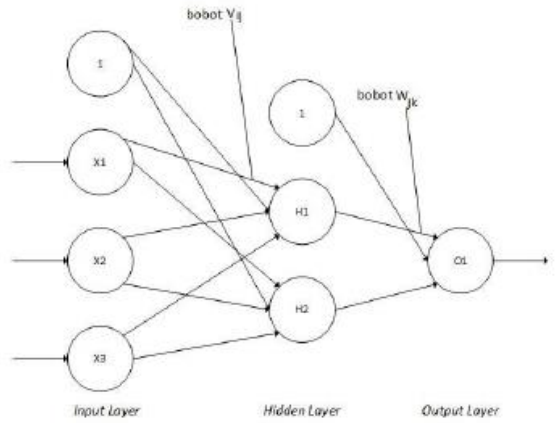
Pada gambar 2.10a terdapat beberapa pattern yang terbagi dalam dua buah class-1 dan class+1. Pattern yang tergabung dalam class-1 dilambangkan dengan kotak warna merah, sedangkan pattern yang ada dalam class+1 dilambangkan dengan lingkaran warna kuning. Masalah klasifikasi dapat diterjemahkan dengan usaha mencari garis (hyperlane) yang memisahkan antara kedua kelompok tersebut. Beberapa alternatif garis pemisah ditampilkan dalam gambar 3a.
Gambar 3b menunjukkan hyperlane yang memisahkan class-1 dengan class+1 yang memiliki margin tertentu. Margin adalah jarak antara hyperlane dengan pattern terdekat dari masing2 class. Pattern terdekat ini disebut sebagai support vecto




Gambar 3. SVM Mencari Hyperlane Terbaik

## 2.4 Backpropagation Neural Network

Backpropagation neural network merupakan tipe jaringan saraf tiruan yang menggunakan metode pembelajaran terawasi (supervised learning). Pada supervised learning terdapat pasangan data input dan output yang dipakai untuk melatih Neural Network hingga diperoleh bobot yang diinginkan. Metode Backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward). Untuk mendapatkan error ini, tahap perambatan maju (forward) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, seperti terlihat pada persamaan 2.29.
Arsitektur Backpropagation Neural Network terdiri atas tiga lapis (layer) yaitu lapis masukan (input layer) xi, lapis tersembunyi (hidden layer) zj, dan lapis keluaran (output keluaran) yk. Lapis masukan dan lapis tersembunyi dihubungkan dengan bobot vij dan antara lapis tersembunyi dan lapis keluaran dihubungkan oleh bobot wjk
Arsitektur dapat dilihat pada gambar 4




Gambar 4 Arsitektur Neural Network.

2.5 Analisis Masalah

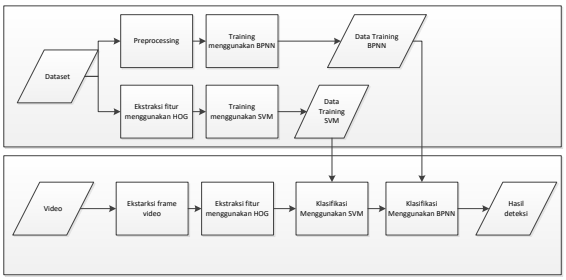
Sistem deteksi merupakan suatu teknologi yang sangat membantu dalam berbagai bidang. Sistem pemantauan lalu lintas yang banyak berkembang selama ini menggunakan magnetic loop detector dan radar sebagai alat pemantau jumlah kendaraan.

Masalah yang didapat pada deteksi mobil menggunakan metode background substraction adalah adanya bayangan yang ditimbulkan akibat cahaya matahari yang mengenai kendaraan akan terdeteksi sebagai objek.

Berdasarkan masalah tersebut maka dibuatlah aplikasi deteksi mobil dengan menggunakan metode histogram of oriented gradient, SVM dan BPNN. Dikarenakan metode SVM dan BPNN merupakan metode yang dapat melakukan klasifikasi objek .

2.6 Analisis Proses

Analisis proses yang akan dilakukan pada penelitian ini dibagi menjadi beberapa tahapan. Pertama adalah tahap training, dimulai dengan ekstraksi fitur pada dataset, kemudian hasil ekstraksi fitur akan dilatih menggunakan metode svm dan bpnn. Setelah tahap training dilakukan tahap testing apakah sistem sudah bisa mengenali mobil atau belum.




Gambar 5 Arsitektur Sistem

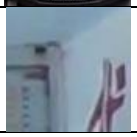
2.7 Analisis Data Masukan

Data masukan yang akan digunakan pada penelitian ini adalah data training dan data uji.

1. Data training

Data trainin yang di gunakan dibagi dalam 2 buah jenis yaitu data training positif dan data training negatif. Data training positif adalah data yang berisikan gambar mobil, sedangkan data training negatif adalah data yang berisikan gambar bukan mobil. Berikut adalah contoh data training positif dan negatif

Tabel 1. Contoh data training

| Gambar | Data |
|---|---|
|  | Positiv |
|  | Positiv |
|  | negativ |

2. Data Uji

Data uji berupa video hasil rekaman kamera statis pada jalan raya. Pada video hasil rekaman tersebut akan dilakukan preprocessing yaitu video dirubah menjadi frameframe untuk proses konversi ke grayscale setelah itu akan masuk ke tahapan ekstraksi fitur mengunakan histogram of oriented gradient. Setelah itu masuk ke tahapan klasifikasi menggunakan SVM dan kemudian klasifikasi menggunakan BPNN.

2.8 Analisis metode

Tahap pertama dalam penelitian ini adalah mengekstrak fitur menggunakan metode HOG dengan melalui tahap:

1. Normalisasi gamma dan warna

Proses Normalize gamma and color atau normalisasi gamma dan warna pada frame gambar dari video terjadi dalam framework HOG pada EmguCV. Normalisasi gamma dan warna pada frame gambar pada sistem ini menggunakan instruksi : void normalize_hists. Proses normalisasi gamma dan warna dari frame gambar pada framework HOG dalam library EmguCV dilakukan pada frame-frame gambar dari video. Instruksi EmguCV ini akan menormalisasikan gamma dan warna pada frame-frame gambar tersebut

2. Compute Gradient

Compute gradients atau perhitungan gradien pada frame citra. Pada tahap kedua ini, citra pertama kali dikonversi kedalam bentuk citra skala keabuan.

Pada contoh kali ini Metode yang digunakan untuk mengkonversi nilai pixel dari citra RGB menjadi grayscale adalah metode Luminosity.

$Y(x,y)=(0.228*R)+(0.518*G)+(0.114*B)$

Berikut adalah contoh gambar dataset mobil:



Gambar 6 contoh citra mobil rgb

Untuk merubah menjadi grayscale nilai-nilai tersebut dihitung menggunakan persamaan sebagai berikut :

Y(1,1) = (0,299*90) + (0,587*94) + ( 0,114*79) = 91,094 ~ 91
Y(1,2) = (0,299*122) + (0,587*126) + (0,114*111) =123,09 ~ 123
Y(1,3) = (0,299*159) + (0,587*163) + (0,114*149) = 155,88 ~ 156
Y(1,4) = (0,299*167) + (0,587*151) + (0,114*147) = 155,33 ~ 155
Y(1,5) = (0,299*165) + (0,587*168) + (0,114*157) = 165,85 ~ 166
Y(1,6) = (0,299*86) + (0,587*79) + (0,114*78) = 86,849 ~ 87
Y(1,7) = (0,299*56) + (0,587*61) + ( 0,114*54) = 58,707 ~ 59
Y(1,8) = (0,299*61) + (0,587*66) + ( 0,114*59) = 63,707 ~ 64
Y(1,9) = (0,299*61) + (0,587*66) + ( 0,114*62) = 64,049 ~ 64
Y(1,10) = (0,299*72) + (0,587*76) + ( 0,114*75) = 74,69 ~ 75
Y(1,11) = (0,299*67) + (0,587*73) + ( 0,114*73) = 71,206 ~ 71
Y(1,12) = (0,299*87) + (0,587*92) + ( 0,114*95) = 90,847 ~ 91
Y(1,13) = (0,299*104) + (0,587*109) + (0,114*115) = 108,189 ~ 108
Y(1,14) = (0,299*88) + (0,587*93) + (0,114*99) = 92,189 ~ 92
Y(1,15) = (0,299*93) + ( 0,587*97) + ( 0,114*106) = 96,83 ~ 97

Setelah perhitungan konversi dari RGB ke grayscale, maka didapatkan citra hasil grayscale sebagai berikut :



Gambar 7 citra hasil grayscale

Setelah citra dikonversi menjadi grayscale, langkah selanjutnya adalah menghitung nilai gradien citra. Metode yang digunakan adalah metode gradien selisih terpusat (center-difference). Secara spesifik metode ini melakukan filtering pada citra grayscale dengan filter kernels dari persamaan berikut:

$$D_2(x) = [-1 \quad 0 \quad 1\,]\ dan\ D_2(y) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Misalkan matriks citra dengan ukuran 3x3 pixel sebagai berikut:

$$i \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \rightarrow \begin{bmatrix} 91 & 123 & 156 \\ 157 & 166 & 182 \\ 155 & 159 & 114 \end{bmatrix}$$

Sebagai contoh akan dihitung nilai konvolusi dari Ix (nilai I pada sumbu x) dan Iy (nilai I pada sumbu y) pada posisi p22 sebagai berikut:

$$Ix(P_{22}) = [P_{21} \quad P_{22} \quad P_{23}] * [-1 \quad 0 \quad 1\,]$$

= [ 157 166 182 ] * [-1 0 1]
= (157 * -1) + (166 * 0) (182 * 1)
= (-157 + 0 + 182)
= 25
Jadi nilai Ix pada pixel adalah 25

$$Iy(P_{22}) = \begin{bmatrix} P_{12} \\ P_{22} \\ P_{32} \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 123 \\ 166 \\ 159 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

= (123 * -1) + (166 * 0) + (159 * 1)
= -123 + 0 + 159
= 36
Jadi nilai Iy pada pixel adalah 36
Dengan didapatnya nilai Ix dan Iy pada koordinat maka dapat dihitung nilai gradient magnitude dan orientasi gradien, Cara menghitung gradien magnitude adalah menggunakan rumus

$$|G| = \sqrt{{I_x}^2 + {I_y}^2}$$

$$G = \sqrt{25^2 + 36^2}$$

$$= 43.82921$$

Maka didapatkan gradien magnitude adalah
43.82821 $\theta = arctan\frac{I_x}{I_y}$
$= \arctan\frac{25}{36} = 35°$
Maka didapatkan nilai orientasi gradien adalah 35°

3. Menghitung Histogram dan Bin Orientasi

Untuk setiap sel akan diakumulasikan histogram 1-D lokal atau orientasi tepi pada semua pixel dalam sel. Kombinasi cell-level histogram 1-D membentuk representasi histogram orientasi dasar. Setiap histogram orientasi membagi berbagai sudut gradient menjadi angka tetap dalam bins yang telah itentukan. Besarnya gradient dari pixel dalam sel digunakan untuk vote ke dalam histogram orientasi. Misalnya akan dibangun histogram yang didistribusikan melalui 0°-180° dengan sejumlah channel sama dengan 9,

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 |

Dan seterusnya Pada tahap menghitung gradien didapatkan gradien magnitude adalah 43.8
dan
gradien orientatiaon adalah 35°

karena 35 berada diantara bin 20 dan 40 maka berikut adalah proses perhitungan menentukan orientasi binnig (Spatial Orientation Binning)

$$\frac{40-35}{20} \times 43.8 = 10.95$$

Nilai diatas didistribusikan ke bin 20°

$$\frac{20-5}{20} \times 43.8 = 32.85$$

Nilai didatas didistribusikan ke bin 40°

| | 10.95 | 32.85 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 20 | 40 | 60 | 80 | 100 | … | … | 160 |

4. Normalisasi Block

Pada saat proses Compute Gradient di peroleh nilai gradien yang berbeda-beda. Sehingga perlu pengelompokkan tiap cells dalam cells perlu dikelompokkan lagi menjadi kelompok yang lebih besar yang disebut block. Sedangkan setelah pengelompokkan menjadi block, block ini biasanya terjadi overlap atau tumpang tindih. Dalam normalisasi block ini menggunakan geometri block persegi R-HOG. Proses ini untuk merupakan proses akhir dari metode HOG yang menghasilkan fitur. Proses ini dilakukan saat proses detector windows seperti pada proses menghitung bin orientasi bin. Ukuran detector windows yang digunakan adalah 64x128 yang terdiri dari 8x8 piksel. Menghitung normalisasi blok yaitu mengambil kelompok sel dan menormalisasi respon kontras secara keseluruhan. Hal ini dilakukan dengan mengakumulasikan ukuran histogram dari grup sel yang disebut blok. Hasilnya akan digunakan untuk menormalisasi setiap sel dalam blok , berikut adalah penghitungan histogram dalam blok.

Misalkan kita memiliki vektor warna RGB [128, 64, 32]. Panjang vektor ini adalah [1282 + 642 + 322} = 146.64. Ini juga disebut norma L2 dari vektor. Membagi setiap elemen vektor ini dengan 146.64 memberi kita vektor yang dinormalisasi [0.87, 0.43, 0.22]. Sekarang perhatikan vektor lain dimana elemen dua kali nilai vektor pertama 2 x [128, 64, 32] = [256, 128, 64]. Anda dapat mengerjakannya sendiri untuk melihat bahwa normalisasi [256, 128, 64] akan menghasilkan [0,87, 0,43, 0,22]

5. Detektor Windows

Langkah terakhir dalam metode HOG adalah mengumpulkan nilai histograms of oriented gradients dalam detector window. Semua histogram untuk setiap block akan dikonkatenasi untuk menghasilkan sebuah nilai feature vector.

pada EmguCV Penghimpunan HOG terhadap jendela deteksi pada frame gambar dari video menggunakan instruksi EmguCV ::gpu::HOGDescriptor::detectMultiScale. Proses penghimpunan HOG terhadap jendela deteksi gambar pada framework HOG dalam library EmguCV dilakukan pada keseluruhan frame gambar dari video. Instruksi EmguCV ini akan melakukan penghimpu nan HOG terhadap jendela deteksi pada gambar-gambar tersebut.

Pada tahap ini akan mengkasilkan suatu feator vector dari gambar yang diproses

0.657589 0.680929 0.641489 0.387971 0.991275 0.178049 0.799521 0.415526 0.651859…….. 0.066560

Hasil dari hog ini akan menjadi inputan dari SVM. Ukuran feature vector dihasilkan dari perkalian dari ukuran block (2x2 cell) , jumlah bin 9, dan banyaknya block yang terbentuk dari image dengan ukuran 128x64. Feature vector inilah yang dipakai sebagai input untuk proses learning SVM. Dalam proses learning dilakukan 2 tahap, yaitu training data dan tes model. Pada training data, feature HOG dari image positif diberi label +1 yang menandakan bahwa image yang dilatih merupakan feature image positif dan feature image negatif diberi label -1 yang menandakan image yang dilatih merupakan feature image negatif. Training SVM ini menghasilkan sebuah model yang akan dites dengan menggunakan classify SVM.

Model hasil learning SVM ini digunakan sebagai acuan dalam mendeteksi mobil. proses deteksi dilakukan dengan melakukan filter dari pojok kiri atas sampai pojok kanan bawah. Apabila dalam proses filter tersebut terdapat bagian image yang sesuai dengan model, maka obyek dianggap sebagai image positif yang berarti merupakan obyek mobil dengan memberi kotak pada obyek tersebut.

Setelah hasil svm keluar sistem akan melakukan crop pada objek yang terdeteksi dan akan melakukan klasifikasi secara backpropagation dengan tahapan.

Objek yang dideteksi menggunakan Histogram dan SVM akan dideteksi menggunakan Backpropagation Neural Network. Dikarenakan metode klasifikasi ini tidak bisa menerima data inputan berupa vektor yang lebih kecil dari -1 dan dara inputan yang lebih besar dari +1 maka data inputan yang dipakai adalah gambar dari objek yang dideteksi menggunakan HOG dan SVM. Kemudian akan dilakukan preprosesing seperti Grayscale, edge detection dan tresholding. Sehingga data yang diinputkan berupa bilangan biner 0 dan 1.

Deteksi tepi

Proses deteksi tepi dijalankan dengan menggunakan metode sobel, metode sobel menggunakan 2 matriks yang berukuran 3x3 yaitu matrix Gx dan matrix Gy kedua matrix tersebut digunakan untuk menghitung perbedaan warna pixel yang sedang dihitung di sekitarnya secara vertikal dan horisontal matrix Gx digunakan untuk menghitung secara horisontal dan matrix Gy digunakan untuk menghitung secara vertikal.

Matrix Gx ditunjukkan pada tabel

Tabel 3.9 Matrix Gx

| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Matrix Gy ditunjukkan pada tabel

Tabel 3.10 Matrix Gy

| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Pada perhitungan deteksi tepi ini semua pixel dihitung kecuali pixel yang berada di tepi gambar, contoh kali ini akan dilakukan perhitungan pada pixel (1,1) dari hasil grayscale pada tabel 3.2 diambil 5 pixel

| X/Y | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 91 | 123 | 156 | 155 |
| 1 | 157 | 166 | 182 | 156 |
| 2 | 155 | 159 | 114 | 49 |
| 3 | 164 | 128 | 50 | 63 |

Berikut adalah perhitungan untuk setiap pixelnya:

Sx= (91 * -1 ) + ( 123 * 0) + (156 * 1) + (157 * -2) + (166 * 0) + (182 * 2) + (155 * -1) + (159 * 0) + (114 * 1) = 74
Sy = (91 * 1 ) + ( 123 * 2) + (156 * 1) + (157 * 0) + (166 * 0) + (182 * 0) + (155 * -1) + (159 * -2) + (114 * -1)= -94
Maka M = |Sx| + |Sy| =74 + 91 =165
Nilai pixel yang baru didapat adalah 165

Tabel 3.11 Nilai Pixel Baru

| X/Y | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | 165 | | |
| 2 | | | | |
| 3 | | | | |

Setelah di dapat nilai pixel yang baru, tahap selanjutnya yang dilakukan adalah tersholding .
Langkah – langkah tresholding adalah

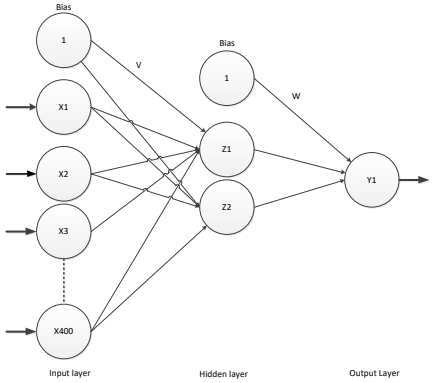
1. Tentukan ambang batas nilai. Dalam contoh ini diambil batas 128
2. Baca setiap pixel gambar, jika nilai pixel gambar lebih besar daripada 128, maka matrix pada pixel yang sama nilai nya diubah menjadi 1, sebaliknya apabila nilai pixel lebih kecil dari pada 128 maka nilai pada pixel matrix yang sama diisi dengan 0.

Contoh hasil tresholding pada pixel (1,1)

Tabel 3.10 Nilai Pixel Hasil Tresholding

| X/Y | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | 1 | | |
| 2 | | | | |
| 3 | | | | |

Setelah dilakukan preprocessing, dilakukan pembelajaran menggunakan backpropagation.
Arsitektur backrpopagation yang digunakan adalah




Gambar 8 arsitektur backpropagation

Yaitu menggunakan 1 hidden layer dengan neuron input sebanyak 400. Didapat dari jumlah pixel yang dideteksi sebagai objek mobil.

2.9 Hasil pegujian.
Pengujian dilakukan pada perangkat lunak dengan membandingkan antara data output dari sistem dan data hasil pengamatan manusia. Pengujian ini dilakukan untuk mengetahui sejauh mana akurasi sistem dapat menghitung jumlah objek mobil yang terekam.
Pengujian dilakukan menngunakan confolution matrix
Skenario pengujian 1
Pada pengujian skenario 1, menggunakan 1 neuron pada hidden layer pada 3 video masing masing berdurasi 4 detik dan 10 detik. Didapatkan hasil

Tabel 4.8 Hasil Deteksi Video Pengujian1.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 3 | 1 | 2 |
| 00:02 | 3 | 1 | 2 |
| 00:03 | 2 | 0 | 2 |
| 00:04 | 2 | 0 | 2 |
| total | 10 | 2 | 8 |

Tabel 4.9 Hasil Deteksi Video Pengujian2.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 7 | 3 | 2 |
| 00:02 | 6 | 4 | 2 |
| 00:03 | 6 | 3 | 1 |
| 00:04 | 5 | 3 | 2 |
| total | 24 | 13 | 7 |

Tabel 4.10 Hasil Deteksi Video Pengujian3.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 5 | 3 | 0 |
| 00:02 | 5 | 6 | 0 |
| 00:03 | 5 | 4 | 0 |
| 00:04 | 5 | 4 | 0 |
| 00:05 | 3 | 4 | 0 |
| 00:06 | 4 | 2 | 1 |
| 00:07 | 4 | 4 | 0 |
| 00:08 | 4 | 4 | 0 |
| 00:09 | 2 | 4 | 0 |
| total | 37 | 35 | 1 |

Tabel 4.11 Confusion Matrix Skenario Pengujian 1

| kelas | | Sistem | |
|---|---|---|---|
| | | mobil | Bukan mobil |
| Aktual | mobil | 10+24+37 | 2+13+35 |
| | Bukan mobil | 1+7+8 | 0 |

Akurasi = $\frac{10+24+37}{10+24+37+2+13+35+1+7+8} * 100\ \% = \frac{71}{137} * 100\% = 51.8\%$

**Skenario Pengujian 2**

Pada pengujian skenario 2, menggunakan 2 neuron pada hidden layer pada 3 video masing masing berdurasi 4 detik dan 10 detik.

Tabel 4.12 Hasil Deteksi Video Pengujian1.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 3 | 1 | 2 |
| 00:02 | 3 | 3 | 2 |
| 00:03 | 2 | 0 | 2 |
| 00:04 | 2 | 0 | 2 |
| total | 10 | 4 | 8 |

Tabel 4.13 Hasil Deteksi Video Pengujian2.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 5 | 3 | 4 |
| 00:02 | 5 | 2 | 3 |
| 00:03 | 6 | 4 | 1 |
| 00:04 | 5 | 4 | 2 |
| total | 21 | 13 | 10 |

Tabel 4.14 Hasil Deteksi Video Pengujian3.avi

| Detik | True Positiv | True negativ | False Positiv |
|---|---|---|---|
| 00:01 | 4 | 2 | 1 |
| 00:02 | 5 | 3 | 0 |
| 00:03 | 4 | 4 | 1 |
| 00:04 | 3 | 3 | 2 |
| 00:05 | 3 | 4 | 0 |
| 00:06 | 3 | 3 | 2 |
| 00:07 | 2 | 2 | 2 |
| 00:08 | 4 | 2 | 0 |
| 00:09 | 2 | 4 | 0 |
| total | 30 | 26 | 6 |

Tabel 4.15 Confusion Matrix Skenario Pengujian 2

| kelas | | Sistem | |
|---|---|---|---|
| | | mobil | Bukan mobil |
| Aktual | mobil | 10+21+30 | 4+13+26 |
| | Bukan mobil | 8+10+6 | 0 |

Akurasi = $\frac{10+21+30}{10+21+30+4+13+26+8+10+6} * 100\ \% = \frac{61}{138} * 100\% = 44.2\%$

Akurasi terbesar yang dicapai dari pengujian performansi ini adalah pada skenario pengujian 1 menggunakan satu hidden layer, dengan hasil akurasi sebesar 51.8%.

## 3. PENUTUP

3.1 Kesimpulan

Berdasarkan dari hasil penelitian, analisis, perancangan sistem, dan implementasi serta pengujian, maka diperoleh kesimpulan bahwa simolator deteksi mobil yang dibangun menggunakan metode histogram of oriented gradients dengan metode klasifikasi SVM dan BPNN memiliki akurasi yang cukup kecil yaitu 51.8% dengan menggunakan 1 neuron pada hidden layer.

3.2 Saran

Berdasarkan hasil penelitian dan implementasi akurasi dari penelitian ini belum mencapai keakuratan yang tinggi. Untuk meningkatkan kinerja dan hasil yang lebih baik, maka diusulkan beberapa saran sebagai berikut :

1. Menggunakan kombinasi metode klasifikasi yang lain seperti naïve bayes dan svm
2. Menggunakan metode ekstraksi fitur yang lain seperti pca.
3. Menggunakan data set yang lebih banyak

## DAFTAR PUSTAKA


[1] Ari Kurniawan Saputra, "Aplikasi Deteksi objek menggunakan Histogram Of Oriented Gradien Untuk Modul Sistem Cerdas Pada Robot NAO," Universitas Bandar Lampung, Bandar Lampung, Skripsi 2016.

[2] Dr. Vinod Shokeen Vandna Singh, "Compariosn of feature Extraction Algorithm For Gender Classification From Face Images," IJERT, vol. 2, no. 5, may 2013.

[3] Cahyo Permata, I Ketut Eddy Purnama, and Muhtadin, "Deteksi Mobil Menggunakan Histogram Of Oriented Gradient," Institut Teknologi Sepuluh November, Surabaya, Skripsi 2010.

[4] K.P Sanak Kumar and Dr.R Bhavani, "Activity Recognition in Egocentic Video Using SVM, kNN and Combined SVMkNN Classifiers," in Internation Conference on Advanced Material Technologies, Visakhapatnam, 2016.

[5] Indu Dokare and Naveeta Kant, "Performance Analysis of SVM, k-NN and BPNN Classifiers for Motor Imagery," International Journal Of Engineering Trends and Technology, vol. 10, pp. 19-23, April 2014.

[6] Wen Zhang, Xijin Tang, and Taketoshi Yoshida, "Text Classification With Support Vector Machine And Backpropagation Neural Network," in ICCS, Berlin, 2007, pp. 150-157.

[7] A. Gunawan, E. Kusnandar and F. E. Gunawan, V. B. Wewengkang, "Visual Traffic Surveillance Using Feature Point Approach," in Procedia Engineering, vol. 50, 2012, pp. 354-360.

[8] D. Zhou, "Bus Passenger Recognition and Track of Video Sequence," Journal of Multimedia, vol. 8, pp. 262-269, Juni 2013.