

Multivariate Gaussian Mixture Model for Preprocessing of Optic Cup Segmentation using Histogram of Oriented Gradient Feature

1st Sudiro,
Department of Electrical and
Information Engineering,
University of Gadjah Mada
Yogyakarta, Indonesia,
SudiroEEN@gmail.com

2nd M Izzarraysadi Wachid
Department of Electrical and
Information Engineering,
University of Gadjah Mada
Yogyakarta, Indonesia,
izzarraysadi@gmail.com

3rd Hanung Adi Nugraha (Advisor)
Department of Electrical and
Information Engineering,
University of Gadjah Mada
Yogyakarta, Indonesia,
adinugroho@ugm.ac.id

4th Eka Legya Frannita (second advisor)
Department of Electrical and
Information Engineering,
University of Gadjah Mada
Yogyakarta, Indonesia,
eka.legya.f@mail.ugm.ac.id

Abstract— Segmentasi Optic Cup sangat penting untuk keperluan biomedis untuk mendiagnosis pasien pengidap glaucoma. Permasalahan yang sering dihadapi pada proses segmentasi adalah pada preprocessing, yakni saat proses blurring untuk menghilangkan pengaruh dari pembuluh darah. Namun pada paper ini, kami justru memanfaatkan adanya pembuluh darah yang melewati optic cup yang memberikan kesan unique untuk optic cup dibandingkan objek lain pada gambar mata.

Penggunaan segmentasi objek berdasarkan warna dengan mengambil kecerahannya, belum robust untuk foto mata dengan titik terang yang menyebar, seperti yang akan dibahas lebih detail nanti. Oleh karena itu, kami memilih menggunakan feature dari objek tersebut, yang menunjukkan ke-unique-an objek tersebut dibanding objek lain. Kami menggunakan Multivariate GMM-EM dengan input berupa fitur HOG dari tiap objek yang kami labeli, sedangkan proses prediksi, menggunakan Sliding window dengan ukuran dan stride window yang bervariasi bergantung pada ukuran gambar. Output dari proses ini adalah roi keberadaan Optic cup, yang kemudian kami proses, sehingga diperoleh optic cup secara bersih.

Keywords—*Optic Cup Segmentation, Multivariate Gaussian Mixture Model, Histogram of Oriented Gradient, Sliding Window.*

I. INTRODUCTION

Optic Cup merupakan struktur penting pada organ mata yang dapat dijadikan acuan untuk proses diagnosis pasien pengidap glaucoma pada bidang kedokteran. Proses diagnosis oleh seorang dokter, tanpa alat bantu komputer diawatirkan terjadi error yang tinggi. Image processing sangat diperlukan untuk mendiagnosis optic cup mata seseorang sedang mengidap/tidaknya penyakit glaucoma dengan presisi, mengingat menggunakan komputer, yang tentunya memiliki ketelitian yang lebih dibandingkan manusia biasa.

Penelitian mengenai proses segmentasi Optic Cup sudah sangat luas dilakukan, namun kebanyakan mengalami kesulitan di bagian pre-processing karena berprinsip menghilangkan kontur pembuluh darah untuk proses segmentasi nya. Penulis menawarkan metode baru dengan memanfaatkan adanya kontur pembuluh darah yang

melintasi Optic Cup. Hal ini karena justru memberikan nilai ke-unique-an yang mencolok dibandingkan objek lain, karena optic cup memiliki intensitas yang paling tinggi dibandingkan objek lain, serta terdapat kontur pembuluh darah yang memiliki intensitas paling rendah, dengan bentuk kontur yang unique juga. Kondisi ini memberikan fitur optic cup yang cukup berbeda secara signifikan dibandingkan fitur objek lainnya, yang nanti nya dapat diklasifikasi menggunakan berbagai jenis classifier. Pemilihan penggunaan fitur ini dilatarbelakangi oleh salah satu sample gambar mata berikut yang memiliki titik terang dengan ukuran kontur yang hampir seukuran dengan optic cup, serta intensitas nya juga sama tingginya:



Pemilihan Multivariate Gaussian Mixture Model - Expectation Maximization (M-GMM-EM) didasarkan pada banyaknya objek pada organ mata yang juga dapat dijadikan acuan diagnosis penyakit mata, meliputi Optic Cup, Optic Disc, Blood Vessel, serta selaput mata, sehingga untuk kedepannya bisa dimanfaatkan untuk segmentasi objek - objek tersebut. Sedangkan untuk segmentasi Optic Cup saja, yang hanya membutuhkan binary classification, cukup menggunakan Support Vector Machine, namun karena untuk kedepannya ada banyak objek tersebut, SVM tidak akan bekerja, sehingga menggunakan M-GMM-EM.

Untuk paper ini akan membahas M-GMM-EM sebagai proses pre-processing saja, untuk proses yang lebih valid, yang tanpa pre-processing dapat langsung men-segmentasi objek - objek Optic Cup, Optic Disc, bahkan Blood vessel dengan presisi, dapat menggunakan Semantic Segmentation, menggunakan network SegNet. Namun pada paper ini hanya membahas pada penggunaan traditional machine learning, dikarenakan waktu yang kurang jika meneliti Semantic Segmentation, serta ketidak-adaannya komputer training. Dan untuk lebih presisi lagi, dibanding Semantic Segmentation, adalah menggunakan Instance Segmentation.

Sedangkan pemilihan Histogram of Oriented Gradient sebagai fitur input, didasarkan pada pemilihan classifier M-GMM-EM yang bersifat Gaussian. Classifier dan data input harus memiliki distribusi yang sama, dengan M-GMM-EM yang bersifat Gaussian, maka fitur input juga harus terdistribusi Gaussian, yang salah satunya yaitu HOG. Walaupun sebenarnya ada metode ekstraksi fitur yang lebih baik, yaitu Local Binary Pattern (LBP), namun metode ini tidak terdistribusi Gaussian, sehingga tidak digunakan. Ketentuan ini berlaku pada classifier multi class, untuk SVM, dapat valid menggunakan HOG maupun LBP.

II. PENGUMPULAN DATA

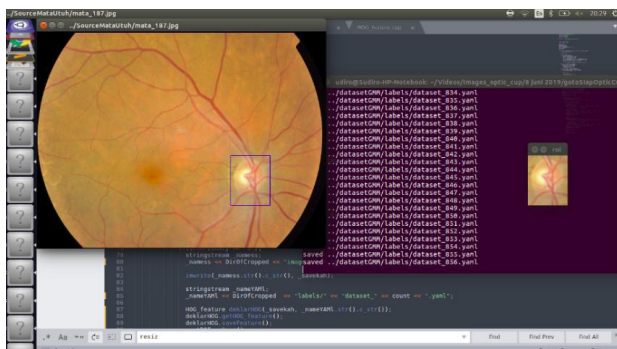
Proses pengumpulan data dapat dilakukan dengan 5 cara yang berbeda:

A. Proses Labeling Secara Offline

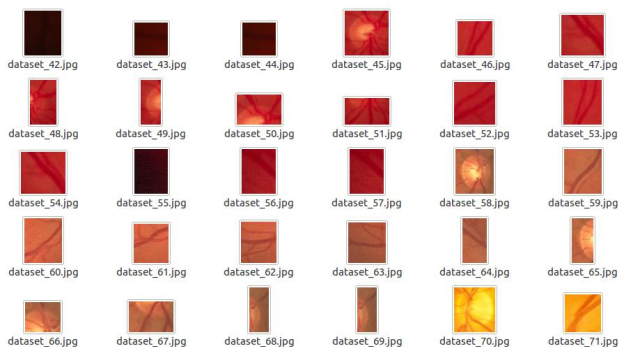
Untuk cara ini, dapat menggunakan tool pelabel gambar, yang kami buat sendiri secara sederhana dengan melabeli objek yang kita inginkan, kemudian diperoleh ROI yang sudah sesuai, dan menyimpannya dengan menekan tombol “s” pada keyboard atau dengan tombol “d” untuk menyimpan sekaligus beralih ke gambar berikutnya pada folder yang dituju.

Data yang disimpan berupa gambar hasil crop serta fitur HOG yang secara langsung diekstrak dari gambar crop tersebut, setelah proses penyimpanan (setelah menekan tombol “s” atau “d”). Untuk uji coba yang telah kami lakukan, dataset fitur yang kami gunakan diperoleh dengan cara ini, dan diperoleh sebanyak 856 buah fitur.

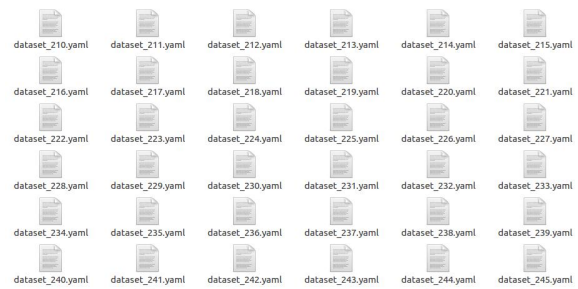
Berikut proses labeling nya:



Sedangkan berikut adalah tampilan dataset gambar hasil pelabelan:



Sedangkan ini adalah tampilan dataset fitur masing - masing hasil crop gambar, yang disimpan ke dalam file YAML:



Dan ini adalah isi dari masing - masing file YAML tersebut, yang berisi matrix 105 x 36, yang pada program, dikonversi menjadi vector 3780x1:

```
1 |YAML:1.0
2 |---
3 |feature: !!opencv-matrix
4 |  rows: 105
5 |  cols: 36
6 |  dt: f
7 |  data: [ 2.64186338e-02, 1.23367552e-02, 1.23821639e-01,
8 | 2.46918738e-01, 2.21533209e-01, 2.46616393e-01, 1.19684063e-01,
9 | 2.20635757e-01, 1.16462782e-01, 3.31602693e-02, 3.76708098e-02,
10 | 1.42978787e-01, 1.21703938e-01, 1.29402936e-01, 1.46370038e-01,
11 | 1.54607356e-01, 2.01881632e-01, 1.39675274e-01, 0.,
12 | 1.66901238e-02, 7.92065337e-02, 1.69062138e-01, 2.24861249e-01,
13 | 1.94621548e-01, 2.56838500e-01, 4.03675705e-01, 1.54457346e-01,
14 | 6.65609201e-04, 4.58994992e-02, 1.32930517e-01, 1.01891071e-01,
15 | 1.94523871e-01, 1.94121063e-01, 1.74424618e-01, 2.31444702e-01,
16 | 1.01236574e-01, 2.64186338e-02, 1.23367552e-02, 1.23821639e-01,
17 | 2.46918738e-01, 2.21533209e-01, 2.46616393e-01, 1.19684063e-01,
18 | 2.20635757e-01, 1.16462782e-01, 3.31602693e-02, 3.76708098e-02,
19 | 1.42978787e-01, 1.21703938e-01, 1.29402936e-01, 1.46370038e-01,
```

- B. Scanning secara otomatis dengan ukuran window dan stride yang bebas ditentukan dari input gambar
- C. Scanning secara otomatis dengan ukuran window dan stride yang bebas ditentukan dari input video dengan stride per frame yang bebas ditentukan.
- D. Scanning secara otomatis dengan ukuran window dan stride yang bebas ditentukan dari input nama folder yang berisikan gambar
- E. Scanning secara otomatis dengan ukuran window dan stride yang bebas ditentukan dari input nama folder yang berisikan video dengan stride per frame yang bebas ditentukan.

Dari kesemua lima cara tersebut, menghasilkan output yang sama, yakni file crop gambar, serta file YAML berisi fitur masing - masing crop tersebut.

III. EKSTRAKSI FITUR HOG

Pada proses ekstraksi fitur nya menggunakan metode Histogram of Oriented Gradient dengan langkah - langkah berikut:

A. Resizing Gambar

Gambar yang akan diekstrak menjadi fitur HOG, pada umumnya akan di-resize menjadi 128x64 yang alasannya akan diketahui pada langkah berikutnya. Namun karena gambar utuh sebelum di-crop memiliki resolusi yang sangat tinggi, walaupun sudah di-crop (pada proses labeling), masih berukuran besar, sehingga jika di-resize menjadi 128x64 dikhawatirkan kehilangan banyak data, oleh karena itu, kami me-resize nya menjadi 192x96 untuk gambar crop yang baris nya lebih besar dari kolomnya, serta 96x192 untuk gambar crop yang barisnya lebih sedikit dibanding kolomnya, untuk menghindari perubahan data jika disamaratakan.

B. Edge Extraction

Dari hasil Resizing tersebut, kemudian diekstrak gradient tiap pixel terhadap pixel sekeliling nya, berupa magnitude dan fase masing - masing gradient, dengan rumus:

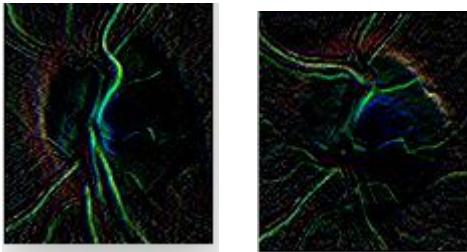
$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

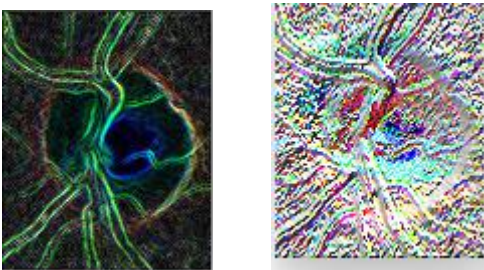
Untuk Gx dan Gy secara berurutan merupakan gradient tiap pixel terhadap neighbor samping dan atas-bawah. Keduanya diperoleh dengan konvolusi dua dimensi antara gambar hasil resize dengan kernel - kernel berikut:

			-1
-1	0	1	0
			1

Dari hasil proses konvolusi tersebut, diperoleh dua buah gambar, yakni hasil gradient terhadap sumbu x dan gradient terhadap sumbu y, dari kiri ke kanan berikut:



Dari Gx (kiri) dan Gy (kanan) dengan rumus sebelumnya, akan diperoleh magnitude dan phase secara berurutan berikut:



Dari nilai magnitude dan phase tersebut, akan diproses pada langkah berikutnya.

Dari gambar yang terdiri dari tiga channel B, G, dan R, hanya akan diambil magnitude edge paling besar di antara tiga tersebut.

C. Partitioning

Dari hasil Gambar Edge tersebut, untuk gambar 192x96, dibagi menjadi 16x8 bagian, dengan tiap bagian merupakan sub gambar yang berukuran 12x6. begitu sebaliknya untuk gambar hasil resize 96x192.

D. Mapping

Dari masing - masing gambar magnitude dan phase yang sudah dipartisi pada langkah (c), kemudian akan di-mapping sesuai posisi pixel nya. Metode mapping untuk HOG ini ada dua jenis, yaitu weighted mapping dan un-weighted mapping.

Pada paper ini kami menggunakan weighted mapping, berikut penjelasannya:

Misalkan gambar magnitude dan phase tersebut diperlihatkan nilainya akan diperoleh berikut:

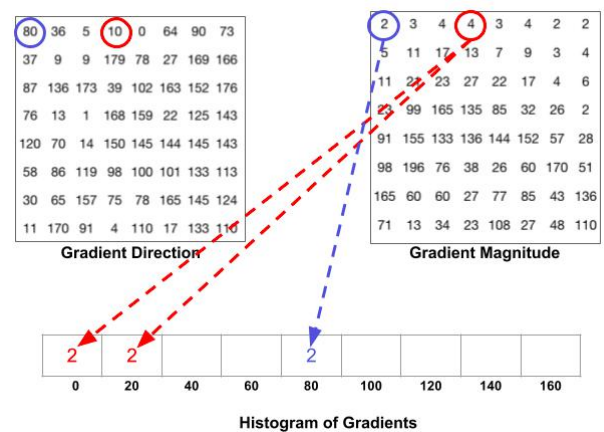
2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

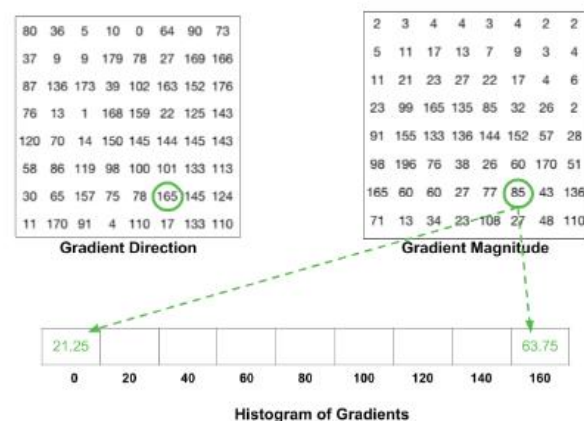
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Dari masing - masing blok nilai tersebut, akan kita mapping dari 0 hingga 180 derajat, dengan pembagian kelas 0 dan 180 masuk kelas 1, 20 masuk kelas 2, 40 masuk kelas 3, 60 masuk kelas 4, 80 masuk kelas 5, 100 masuk kelas 6, 120 masuk kelas 7, 140 masuk kelas 8, dan 160 masuk kelas 9, sehingga keseluruhan terdapat 9 buah kelas.



Kemudian untuk nilai sudut phase yang berada di antara nilai sudut masing - masing kelas, akan dibagi persentasi secara reciprocal untuk kelas bawah dan kelas atas nya. Misalkan untuk magnitude 85 diperoleh phase 165 pada gambar berikut:



Nilai 165 berada di antara 160 (kelas 9) dan 180 (kelas 1), sehingga untuk kelas 9 akan memperoleh bagian:

$$(180 - 165)/20 * 85 = 63.75$$

Sedangkan untuk kelas 1, diperoleh:

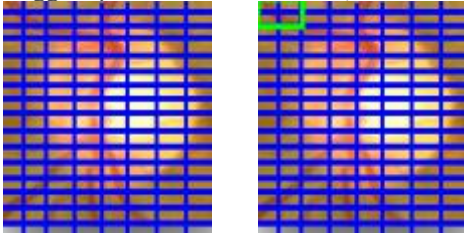
$$(165 - 160)/20 * 85 = 21.25$$

Perhitungan di atas benar, **tidak** salah, karena perhitungan secara reciprocal didasarkan pada seberapa dekat nilai phase tersebut terhadap nilai masing - masing kelas.

Nilai yang diperoleh pada masing - masing kelas merupakan kontribusi dari magnitude tiap pixel yang ditambahkan secara erus menerus untuk semua pixel di sub-image 12x6 atau 6x12 tersebut.

E. Normalization

Untuk masing - masing bagian pada gambar terkadang merupakan objek yang sama namun karena terkena pancaran cahaya dengan intensitas yang berbeda, sehingga nilai gradient nya bisa beda. Untuk menghindari kondisi tersebut, maka kami normalisasi tiap empat buah sub-image yang masing - masing terdiri dari 9 kelas (yang disusun dalam satu vektor dengan 9 baris) digabung menjadi satu, sehingga diperoleh vektor 36 baris, berikut ilustrasi nya:



Kiri adalah proses partitioning, dan kanan adalah proses normalisasi.

Pada tiap normalisasi diperoleh vektor 36 baris, sedangkan tiap frame terdapat 7 kali normalisasi mendatar dan 15 kali normalisasi menurun, sehingga dalam satu frame dihasilkan feature HOG berupa vektor berukuran $36 * 7 * 15 = 3780$ baris.

IV. MULTIVARIATE GAUSSIAN MIXTURE MODEL - EXPECTATION MAXIMIZATION

Gaussian Mixture Model merupakan salah satu traditional machine learning yang termasuk kelas unsupervised learning, yakni kita tidak perlu melabeli tiap dataset berdasarkan kluster yang diinginkan, karena GMM akan dengan sendirinya mengklusterisasi dataset yang ada. Sedangkan proses pelabelan yang dijelaskan di awal bukanlah pelabelan data berdasarkan kluster yang diinginkan, namun hanya pelabelan data yang diinginkan dari frame gambar yang tersedia untuk kemudian gambar crop tersebut dicari fitur HOG nya.

Karena dengan menggunakan fitur HOG, maka setiap objek diwakili oleh 3780 angka, atau dengan kata lain setiap objek memiliki 3780 parameter tertentu. Sehingga metode GMM yang digunakan adalah Multivariate GMM (M-GMM).

Inti dari metode GMM ini adalah didasarkan pada seberapa besar peluang tiap data termasuk dalam tiap kluster yang dihitung dengan perhitungan probability density function nya terhadap semua dataset yang ada, serta peluang tiap kluster terhadap semua kluster yang ada. Berikut rumus perhitungannya:

$$p_k(\mathbf{x}|\theta_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1} (\mathbf{x}-\mu_k)}$$

$$\theta_k = \{\mu_k, \Sigma_k\}.$$

Yang menunjukkan probability density function keanggotaan fitur \mathbf{x} pada kluster k terhadap semua dataset fitur yang ada. Dengan μ_k adalah rerata tiap rerata, Σ_k covariance

matrix kluster k , dan d adalah dimensi masing - masing fitur data, yakni 3780.

Dari perhitungan probability density function tersebut, akan diproses secara iteratif mulai dari perhitungan nilai initial dengan metode Expectation Maximization.

Sebelum menjelaskan lebih detail, parameter - parameter yang digunakan pada algoritma M-GMM-EM ini meliputi:

1. Mean masing - masing kluster μ_k ,
2. Covariance matrix Σ_k ,
3. Peluang atau probability density function (pdf) keanggotaan tiap data dari dataset pada tiap kluster p ,
4. Bobot keanggotaan masing - masing data dari dataset pada masing - masing kluster w^k ,
5. Bobot masing - masing kluster terhadap keseluruhan kluster α_k .

E-step

Untuk menginisialisasi μ_k dapat dengan nilai sembarang, karena algoritma ini akan sendirinya mengklusterisasi dataset menjadi beberapa bagian sebanyak kluster yang didefinisikan di awal. Untuk mudahnya, kita definisikan nilainya sebagai:

$$\mu_k = [1 \ 1 \ \dots \ 1]^T_d$$

Langkah berikutnya yaitu dengan menginisialisasi bobot tiap kluster α_k yakni dengan rumus:

$$\alpha_k = \frac{1}{K}$$

Dengan N adalah banyaknya kluster yang didefinisikan.

Dari dua nilai ini, kemudian kita dapat tentukan nilai parameter lain dengan formula berikut:

$$w_{ik} = \frac{\alpha_k p_{ik}}{\sum_{k=1}^K \alpha_k p_{ik}}, 1 \leq k \leq K, 1 \leq i \leq N$$

Bobot keanggotaan ini juga disebut sebagai *responsibility*, dengan K adalah banyak nya kluster, N adalah banyaknya dataset, kemudian p_{ik} diperoleh menggunakan rumus di awal, yang di dalam nya juga terdapat covariance matrix, yang dapat dicari dengan formula:

$$\Sigma_k = \frac{(x-\mu_k)(x-\mu_k)^T}{N}$$

Nilai *responsibility* dibatasi dengan *responsibility* keanggotaan masing - masing data untuk setiap kluster, jika dijumlahkan, maksimal harus 1, atau dengan kata lain:

$$\sum_{k=1}^K w_{ik} = 1$$

Kemudian setelah mendapatkan semua nilai inisial, langkah berikutnya adalah ke Maximization:

M-step:

Mencari nilai N_k yang merupakan penjumlahan dari semua peluang keanggotaan seluruh data pada dataset terhadap kluster tertentu:

$$N_k = \sum_{i=1}^N w_{ik}$$

Nilai ini digunakan untuk meng-update nilai α_k menjadi:

$$\alpha_k^{new} = \frac{N_k}{N}, \quad 1 \leq k \leq K.$$

Selain itu diperoleh juga nilai update dari rerata menjadi:

$$\mu_k^{new} = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} \cdot x_i \quad 1 \leq k \leq K.$$

Dan untuk covariance matrix nya adalah:

$$\Sigma_k^{new} = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} \cdot (x_i - \mu_k^{new})(x_i - \mu_k^{new})^t \quad 1 \leq k \leq K.$$

Begitu seterusnya hingga mencapai kondisi konvergence, yang ditandai oleh nilai Log-Likely-Hood nya sudah kurang dari batas minimal yang ditentukan (untuk kasus ini, kami mengeset batas minimal nya sebesar 10^{-8}).

Sedangkan Log-Likely-Hood sendiri didefinisikan sebagai:

$$\log l(\Theta) = \sum_{i=1}^N \log p(x_i | \Theta) = \sum_{i=1}^N \left(\log \sum_{k=1}^K \alpha_k p_k(x_i | z_k, \theta_k) \right)$$

Setelah mencapai konvergence, maka hasil training yang disimpan adalah berupa Mean, covariance matrix, serta bobot masing - masing klaster, yang kami simpan dalam file YAML seperti hal nya data - data yang lain.

NOTE: covariance matrix seringkali berupa mtrix singular, sehingga gagal dalam perhitungan inverse, untuk mengantisipasi hal tersebut, dapat dilakukan dengan menambahkan factor noise (nilai yang cukup kecil) pada bagian diagonal matrix covariance tersebut, sampai tidak singular lagi.

V. SLIDING WINDOW SEARCHING

Setelah mendapatkan mean, covariance matrix, serta bobot setiap klaster dari hasil training, proses selanjutnya adalah memprediksi lokasi objek optic cup pada gambar 'asing'. Karena optic cup merupakan objek yang tentunya tidak hanya terdiri dari satu pixel, maka langkah nya yaitu kita mengambil sampling bounding box dari gambar utuh untuk kemudian diekstrak fitur HOG nya, dan dihitung seberapa besar bobot dan peluang keanggotaan tiap bounding box tersebut.

Di awal, sudah kita jelaskan bahwa karakteristik dari Optic Cup berbentuk kontur tertutup (menyerupai lingkaran), memiliki intensitas yang lebih tinggi dibandingkan objek lainnya, serta dilalui oleh pembuluh darah dengan intensitas yang cukup rendah, serta bentuk kontur yang unique. Namun ciri - ciri ini hampir mirip jika kita melihat objek mata pada bagian pinggiran (yang bersampingan dengan background hitam) yang bentuk nya lengkung, serta ada kemungkinan terdapat pembuluh darah di sana, sehingga dikhawatirkan tidak valid, jika bagian tersebut diikuti dalam proses prediksi. Sehingga proses prediksi, dengan pengambilan bounding box gambar mentah hanya dilakukan pada bagian gambar yang intensitas nya di atas threshold (sedemikian sehingga bagian hitam pada gambar mentah mata tidak ikut).

Dari data - data bounding box yang sudah diperoleh, kemudian diekstrak fitur HOG nya, dan dihitung bobot dan peluang masing - masing fitur terhadap klaster - klaster yang ada. Kemudian korespondensi klaster yang memiliki

peluang dan bobot terbesar untuk suatu fitur menunjukkan bahwa fitur tersebut termasuk dalam klaster tersebut.

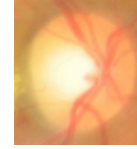
Nilai yang dihitung adalah

$$\alpha_k p_{ik}$$

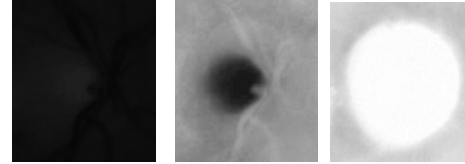
Dan dari nilai maksimum dari kesemua klaster yang ada.

VI. AFTER PREPROCESSING

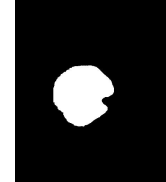
Dari hasil deteksi berupa crop gambar berikut:



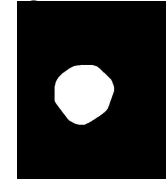
Kemudian kita konvert ke HSV, kemudian kita split menjadi masing - masing H, S, dan V channel, secara berurutan dari kiri ke kanan diperoleh:



Dari ketiga gambar tersebut, yang paling mencolok bagian pembuluh darahnya adalah pada channel S, untuk membersihkan nya lagi, kita akan threshold binary invert dengan batas threshold sebesar 220, sehingga diperoleh:



Kemudian kita cari kontur terluar dari gambar di atas, namun yang closed loop (untuk mengantisipasi pixel putih yang tidak terlihat kasat mata oleh kita), serta dengan ukuran yang melebihi batas minimal, kemudian dari kontur - kontur tersebut dijadikan satu, kemudian dilakukan convexHull, sehingga diperoleh hasil berikut:



Dari sini, kita sudah peroleh segmentasi dari Optic Cup.

FUTURE WORK

Pada saat training sering kali terjadi divergensi pada GMM yang disebabkan oleh dataset yang digunakan. Untuk kedepannya mungkin dapat dicari algoritma bantuan untuk menentukan algoritma yang sesuai untuk mencari dataset yang bagus, sehingga tidak terjadi divergensi, karena saat terjadi divergensi, troubleshooting nya sangat tidak nyaman, dikarenakan tiap step membutuhkan lebih dari 30 menit, jadi harus menunggu selama waktu tersebut untuk mengetahui apakah divergen atau konvergen.

Hasil convexHull dari hasil akhir di atas terlihat terlalu convex, padahal diharapkan hanya bagian lengkung (bagian kanan saja) yang di convex, sedangkan saat menggunakan approxisasi kontur hasilnya tidak sampai ter-convex, akan diperbaiki untuk kedepannya.

REFERENCES

Referensi yang kami gunakan, hanya pada penjelasan teoritis serta searching secara random di internet, serta

terinspirasi dari hasil color segmentation menggunakan GMM-EM, dan kami aplikasikan untuk fitur suatu objek.

- [1] Fu. Zhaoxia, Wang. Liming, "Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm" Phil. China.