

Perbandingan Performa *Pre-Trained Classifier dLib* dan *HAAR Cascade (OpenCV)* Untuk Mendeteksi Wajah

Nanda Wahyudiana^{#1}, Setia Budi^{*2}

#Teknologi Informasi, Universitas Kristen Maranatha
Jl. Prof. drg. Surya Sumantri, No.65, Bandung

¹nandawahyudiana@gmail.com

²setia.budi@it.maranatha.edu

Abstract — Face detection is a part of Computer Vision and subsection of the detection of objects. Computer Vision is defined as the branch of science that studies how the computer can recognize an object. Face detection is not yet implemented on the learning environment. Currently, the attendace process is still using paper and cheating can occur. In this final project will compare the two pre-trained classifier applied to the learning environment by using datasets IBAtS. Two pre-trained classifier to compare performance is haar cascade and dlib. To find the best classifier then two classifier evaluation is done in order to better compareable. From the results of the evaluations already carried out have been inferred that haar cascade is a good classifier for applied learning environment.

Keywords— Computer Vision, dLib, Face Detection, Haar Cascade, IBAtS

I. PENDAHULUAN

Deteksi wajah merupakan bagian dari Computer Vision dan sub-bagian dari deteksi objek. Deteksi wajah merupakan tahap awal dari semua metode analisis wajah seperti pengenalan wajah, penyalarsan wajah, pemodelan wajah, verifikasi wajah, dan pelacakan wajah. Pendahuluan memuat identifikasi masalah, tinjauan pustaka, tujuan penelitian, serta manfaat yang didapatkan dari hasil penelitian. Dasar tujuan utama dari algoritma deteksi wajah adalah menentukan adanya wajah dalam gambar atau tidak. Tantangan utama dalam deteksi wajah adalah keragaman pose wajah dan ekspresi wajah, objek penghalang, kondisi pencahayaan yang berbeda-beda, variasi skala gambar, warna, dan tekstur [1]. Paragraf dimulai menjorok ke dalam seperti dalam contoh paragraf ini.

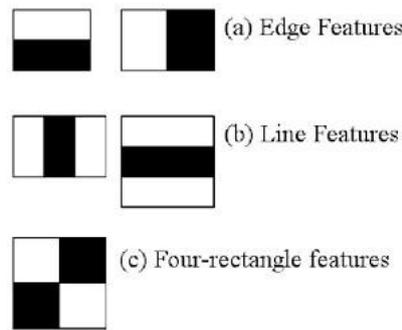
Metode deteksi wajah dapat dibagi ke dalam empat kategori, yaitu: Knowledge-Based [2], Feature-Based [3], Template Matching, Appearance-Based [4] [5]. Knowledge-Based merupakan metode yang menggunakan pengetahuan dari manusia mengenai tipikal wajah manusia. Biasanya, metode ini mengambil hubungan antara fitur wajah yang diberi oleh pengguna tersebut. Feature-Based merupakan algoritma yang berujuan untuk mencari struktural features yang telah ada walaupun kondisi dari gambar bervariasi (pose, pencahayaan, dan posisi gambar). Template Matching Method merupakan beberapa standar pola wajah disimpan untuk menggambarkan wajah sebagai keseluruhan atau facial features secara terpisah. Korelasi antara gambar masukan dan pola disimpan dan dihitung untuk deteksi. Appearance-Based berbanding terbalik dengan Template Matching model atau template belajar dari gambar training di mana sudah terdapat variabilitas dari bentuk wajah. Dari 4 pendekatan tersebut, Appearance-Based mempunyai performa yang lebih baik dari pendekatan lainnya. Secara umum metode Appearance-Based sangat bergantung pada analisis statistik dan machine learning untuk menemukan karakteristik wajah dari gambar [6].

II. KAJIAN LITERATUR

A. HAAR Cascad

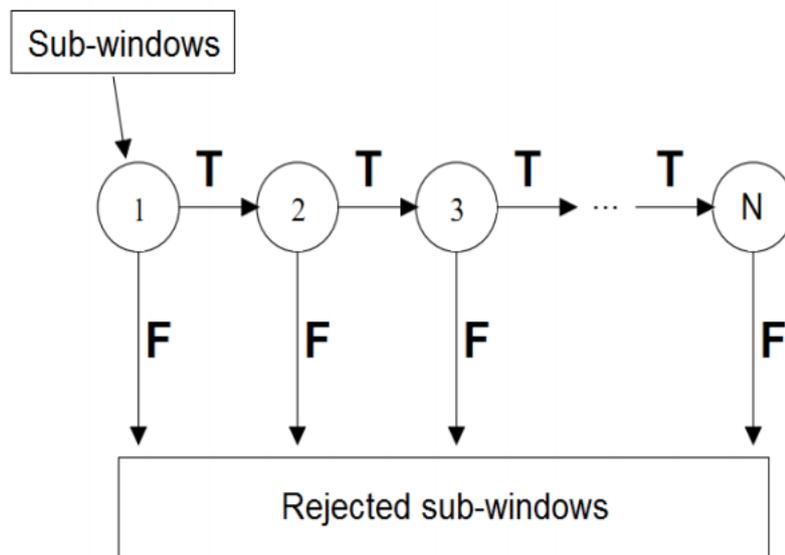
Deteksi wajah telah ditingkatkan dalam hal kecepatan dengan mengaplikasikan *haar-feature* dengan kontribusi kerangka deteksi objek ViolaJones [7]. Implementasi dari kerangka ini, seperti *OpenCV*, menyediakan berbagai macam *face classifier* dibuat oleh beberapa peneliti dan menggunakan dataset yang berbeda untuk dipelajari. Performa dan kehandalan dari classifier yang ada sangat beragam.

Viola et al. [7] mengajukan kerangka deteksi objek menggunakan *Haar-Like Feature*, dimana sudah banyak digunakan bukan hanya untuk deteksi wajah, namun untuk deteksi objek juga.



Gambar 1 Haar Like Feature

Haar-Like Feature mempunyai 4 bentuk dalam mendeteksi wajah yaitu : (a) *Edge Features*, (b) *Line Features*, dan (c) *Four-Rectangle features*. *Edge Features* terbagi menjadi dua warna yaitu hitam dan putih dan berbentuk horizontal dan vertikal yang berguna untuk mendeteksi *Edges*. *Line Features* mempunyai tiga warna yaitu putih, hitam dan putih (dapat dibalik menjadi hitam, putih, hitam) dan juga mempunyai dua bentuk yaitu vertikal dan horizontal yang mempunyai fungsi untuk mendeteksi *Line*. *Four-Rectangle features* merupakan *special features* yang menghitung perbedaan jumlah nilai variasi dari *edge* dan *line features*. Semua *features* tersebut dipakai untuk mendeteksi fitur wajah yang relevan.



Gambar 2 Sub-Windows

Berkat implementasi *Open Computer Vision Library (OpenCV)* [8], kerangka deteksi objek menjadi sangat populer dan memotivasi komunitas untuk membuat deteksi objek mereka sendiri. *Classifier* ini memakai *haar-like feature* yang diterapkan pada sebuah gambar. Hanya daerah pada gambar itu, yang biasa disebut *sub-windows*, yang dikirimkan pada *classifier* untuk dideteksi yang dianggap objek oleh detektor. Gambar 2 menunjukkan skema *detection cascade* dengan *N* stages. *Detection cascade* didesain untuk meng-eliminasi angka besar dari *negative example* dengan sedikit proses [9].

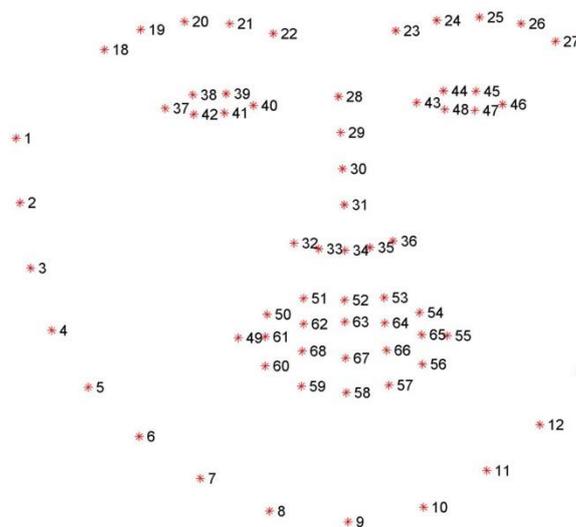
B. dLib

Dlib merupakan sebuah *library machine learning* yang ditulis dalam bahasa pemrograman C++ dan ditunjukkan untuk menyelesaikan masalah kehidupan sehari-hari. Dlib adalah *library open source* yang biasa dipakai untuk pengembangan *software machine learning*. Inti dari dlib adalah sebuah aljabar linier dengan *Basic Linear Algebra Subprograms (BLAS)*. *BLAS* biasa digunakan untuk implementasi *bayesian network* dan algoritma *kernel-based* untuk klasifikasi, *clustering*, deteksi anomali, regresi dan *feature ranking*. Dlib mempunyai dua komponen penting yaitu aljabar linier dan *machine learning tools* [10].

Komponen aljabar linier berbasis pada teknik *template expression* yang ada pada Blitz++. Blitz++ merupakan sebuah *numerical software* yang dikembangkan oleh Veldhuizen dan Ponnambalam (1996). Dlib menggunakan *BLAS* agar mendapatkan performa terbaik dan menambah kecepatan sebagai sebuah *library*. Dlib dapat menunjukkan semua transformasi pada semua ekspresi dengan melibatkan *BLAS* yang dapat digunakan. Lalu *BLAS* dapat membuat pengguna melakukan penulisan atau penjumlahan dalam bentuk intuisi dan diberikan pada *library* [10].

Tujuan utama pada *machine learning tools* adalah untuk menyajikan sebuah arsitektur modular dan mudah digunakan. dLib dapat diimplementasikan pada gambar, kolom vektor atau struktur bentuk data. Implementasi dari algoritma akan berbeda pada data dengan apa yang akan dioperasikan nantinya. Karena dlib sangat fleksibel maka dapat dioperasikan secara langsung pada objek apapun dimana itu membuat kernel menjadi custom dimana biasanya kernel hanya bisa pada objek yang sudah jelas ukurannya [10].

Facial Landmark digunakan oleh dLib untuk mendeteksi wajah manusia. dLib menggunakan *training set* yang sudah ditandai (*facial landmark*).



Gambar 3 68 Titik Facial Landmark

Terdapat 68 titik wajah di dalam training set yang dipakai oleh dLib seperti pada Gambar. Titik tersebut ditandai secara spesifik dengan koordinat x dan y. Dengan menggunakan dataset dari iBUG 300-W dataset sehingga dLib dapat mendeteksi wajah [11].

C. Confusion Matrix

Confusion matrix merupakan suatu kesimpulan klasifikasi dari sebuah *classifier*. *Confusion matrix* pada umumnya memiliki dua buah ruang dimensi, satu sisi menunjukkan nilai sesuai kenyataan dan satu dimensi lainnya mewakili nilai dari hasil prediksi. Dari hasil confusion matrix yang diperoleh, akan dicari beberapa nilai, antara lain nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) dari setiap kelas atau label yang ada. Suatu nilai dapat dikatakan sebagai TP apabila nilai kenyataan bernilai benar dan nilai terprediksi sebagai benar. Suatu nilai dapat dikatakan sebagai TN apabila nilai kenyataan bernilai salah dan nilai terprediksi sebagai salah. Suatu nilai dapat dikatakan sebagai FP apabila nilai kenyataan bernilai salah, tetapi nilai terprediksi sebagai benar. Suatu nilai dapat dikatakan sebagai FN apabila nilai kenyataan bernilai benar tetapi nilai terprediksi sebagai salah.

Nilai TN, TP, FN, dan FP dapat digunakan untuk mencari nilai accuracy, precision dan sensitivity dari masing masing kelas. Accuracy merupakan nilai yang mengukur tingkat kesuksesan dari classifier dalam mengklasifikasikan data dengan benar yang dapat dihitung dengan menjumlahkan nilai TP dan TN yang kemudian dibagi dengan jumlah total nilai TP, TN, FP, FN dan bila ingin dikeluarkan dalam bentuk persentase maka hanya dikalikan 100. Nilai precision dapat dihitung dengan membagi nilai TP dibagi dengan jumlah nilai TP dan FP. Nilai sensitivity dapat dihitung dengan membagi nilai TP dengan jumlah nilai TP dan FN. Bila sudah mendapatkan nilai dari precision dan sensitivity maka dapat didapat sebuah F1-Score. F1-score adalah rata-rata dari precision dan sensitivity. F1-Score yang biasa dipakai adalah F1-Score [12] [13] [14].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

$$F1 - Score = 2 \times \frac{Precision * Sensitivity}{Precision + Sensitivity} \quad (4)$$

III. METODOLOGI

A. Dataset

Dataset memiliki peran penting, karena untuk memastikan dan mendapatkan performa yang terbaik dari kedua metode yang akan dibandingkan yaitu *dlib* dan *haar cascade*. Penerapan kedua metode deteksi wajah akan dilakukan pada lingkungan pembelajaran.

IBaTS (*Image Based Attendance System*) adalah sebuah sistem presensi kelas yang berbasis foto dari mahasiswa yang sedang mengikuti kegiatan belajar. Tujuan dari penelitian ini adalah untuk membuat presensi digital dan mengurangi penggunaan kertas. *Dataset* yang dibangun merupakan lingkungan belajar yang ada di Laboratorium Fakultas Teknik Informasi Universitas Kristen Maranatha. IBaTS Memperoleh data dengan mengambil gambar mahasiswa yang sedang duduk di ruangan Laboratorium Fakultas Teknologi Informasi Universitas Kristen Maranatha. Setiap Laboratorium FIT dapat menampung hingga 40 mahasiswa. Lima kelas dengan kode kelas yang berbeda, berada dalam

observasi untuk pengumpulan data [4]. Data berbentuk gambar(foto) diambil setiap minggu selama 11 minggu berturut-turut. Pengambilan data gambar(foto) memakai smartphones yaitu *blackberry passport* dan *asus zenfone 2 selfie*.

Sebelum mengambil data berbentuk gambar(foto), consent form dibagikan untuk mencari partisipan yang ingin berpartisipasi dalam pengumpulan data. Jika tidak mengisi *consent form* maka wajah mahasiswa tersebut akan diberi kotak berwarna merah untuk menutupi wajahnya.

Gambar tersebut berukuran 1024 pixels dan telah dimasukkan ke dalam direktori kode kelas dan tanggal pada saat foto tersebut diambil. Dalam direktori tersebut juga terdapat Region Of Interest (ROI) yang berbentuk square atau kotak, ROI tersebut menempel pada wajah dari para mahasiswa sebagai tanda bahwa di dalam kotak tersebut merupakan wajah dari manusia. Semua ROI yang ada di dalam sebuah gambar pada dataset tersebut disebut dengan gold standard seperti pada Gambar.

Dalam dataset IBAtS terdapat sebuah tanda wajah yang diberi nama Gold Standard. Gold standard adalah sebuah tag (ROI berbentuk persegi atau square) yang telah melalui proses manual tagging [28]. Gold standard tersebut ditulis dalam format *json*.



a. Contoh Dataset Yang Digunakan



b. Contoh ROI Gold Standard

Gambar 4 Contoh Dataset

Pembersihan data adalah proses mengoreksi (atau menghapus) suatu data yang tidak akurat, lalu memodifikasi, mengganti atau menghapus data yang kotor atau kasar. Pembersihan data dapat dilakukan dalam 2 cara yaitu dengan menggunakan aplikasi (*tools*) dan dengan cara memperbaiki sendiri secara manual. Pada tahap ini, peneliti akan memperbaiki data dengan cara manual yaitu dengan membuka *file json* data yang kotor lalu mengoreksi nilai yang dianggap kurang tepat.

Dataset yang tersedia, perlu diperiksa kembali agar data yang dihasilkan sangat tepat dan akurat. Terdapat 5 buah *instance* yang ada di dalam *Gold Standard* yaitu *x*, *y*, *width*, *height* dan *color*. Gold standard yang sudah disimpan dalam bentuk json perlu ditinjau kembali perihal titik *x*, *y*, ukuran *weight* dan *height* guna mencapai hasil yang lebih baik. Pada Gambar 4 terlihat beberapa kesalahan pada *gold standard* yang perlu dikoreksi seperti beberapa kesalahan menandai objek dan wajah yang tidak ditandai.

Dua cara untuk melakukan proses pembersihan *dataset* yaitu:

- Menggunakan deteksi wajah dari *dlib* terhadap *dataset* dan *gold standard* yang ada. Jika ada titik yang kurang tepat (sehingga mengganggu proses perhitungan evaluasi) maka titik tersebut perlu digeser atau diganti sesuai dengan titik yang dihasilkan oleh *dlib*.
- Melakukan koreksi pada *file json gold standard*.



False Positive (FP)



False Negative (FN)



False Positive (FP)

Gambar 5 Pembersihan Dataset

B. Experimental Setup

Penelitian ini berfokus pada komparasi *pre-trained classifier* deteksi wajah yang tersedia bebas secara *online* yaitu *OpenCV* dan *dlib*. Pada *library OpenCV* terdapat 4 buah nama *classifier* yaitu : *alt*, *al2*, *tre* dan *def*. Ke-empat model tersebut merupakan *haar-cascade* dengan *data training* yang berbeda. Pada penelitian ini, peneliti akan memakai *alt* karena menurut [28] *alt* lebih superior bila diterapkan pada lingkungan belajar. Kedua model tersebut akan diaplikasikan pada semua *dataset* yang telah ditentukan. Pada proses ini, semua nilai *output* yang dikeluarkan akan diubah menjadi objek dan ditampung dalam sebuah file berbentuk *json*. Masing-masing *file* gambar akan memiliki dua buah *file json* yang berisi nilai *output* dari masing-masing model. Untuk memudahkan proses ini, peneliti menggunakan sebuah program untuk menyimpan nilai *output* ke dalam bentuk *json*.

Json yang sudah diisi oleh objek data nilai *output* tersebut disimpan untuk maju ke tahap selanjutnya yaitu visualisasi. Pada tahap visualisasi, peneliti akan menggunakan *jupyter notebook* sebagai media visualisasi gambar *dataset* yang telah dideteksi oleh kedua model. *File json* yang ada akan di-load menggunakan *library json* untuk menandai wajah yang terdeteksi oleh masing-masing model. Deteksi tersebut berupa bentuk kotak (*square*) yang telah dibedakan warnanya agar sebagai pembeda masing-masing *pre-trained classifier*, contoh : *dlib* menggunakan warna merah sedangkan *haar cascade* menggunakan warna biru. Tahap ini akan diulang pada masing-masing ukuran gambar *dataset* yaitu : 100%, 75%, 50% dan 25%.

Untuk mencapai tahap perbandingan dan evaluasi, perlu dilakukan sebuah standarisasi output dari kedua metode tersebut. Kedua metode pre-trained classifier mengeluarkan bentuk output yang berbeda sehingga menjadi tidak comparable jika akan dibandingkan performanya.

Haar Cascade Classifier mengeluarkan nilai berupa (x,y,width dan height) dimana x, y, width dan height adalah sebuah koordinat yang membentuk ROI. Nilai width dan height yang dikeluarkan dari metode tersebut bernilai sama, sehingga bentuk yang dikeluarkan pada ROI adalah square atau kotak.

Face detection dari dlib mengeluarkan nilai berupa berupa (top, bottom, left, dan right) dengan hasil output sudah dijumlahkan dengan panjang dan lebar yang sama sehingga berbentuk square atau kotak.

Standarisasi yang akan dilakukan ialah mengubah nilai output agar kedua metode pre-trained classifier di atas mengeluarkan jenis output yang sama. Peneliti memutuskan untuk melakukan standarisasi nilai output ke dalam format yang sama dengan gold standard dataset dengan 5 buah instance yaitu x, y, width, height dan color. Bila tidak dilakukan standarisasi maka akan terasa tidak fair karena nilai output yang dihasilkan berbeda dengan nilai yang disimpan pada gold standard dataset.



Gambar 6 Dataset Harus Di Standarisasi

Gambar 6 merupakan beberapa contoh deteksi wajah yang harus distandarisasi. Kotak berwarna hijau adalah gold standard dan kotak berwarna merah adalah deteksi wajah dari dlib. Terlihat pada Gambar 6 kotak dlib lebih kecil dibandingkan dengan gold standard, maka dari itu gold standard pun harus diubah nilai pada json file agar menjadi lebih baik.

Kedua metode pre-trained classifier akan dievaluasi untuk diterapkan pada lingkungan edukasi di Laboratorium Fakultas Teknik Informasi Universitas Kristen Maranatha agar data yang dihasilkan dapat dimanfaatkan dengan baik dan cukup akurat untuk digunakan sebagai bahan perbandingan.

Klasifikasi performa terbaik dari kedua model dapat dinilai dengan menggunakan metode confusion matrix, dengan peneliti ingin mengetahui performa terbaik dari masing-masing model untuk diterapkan dalam lingkungan belajar. Dari confusion matrix maka akan memperoleh nilai True Positive (TP), False Negative (FN), dan False Positive (FP).

Nilai TP merupakan hasil prediksi yang dianggap benar (wajah) dan yang dideteksi merupakan wajah. Nilai FN merupakan hasil prediksi klasifikasi salah (tidak ada wajah) dan seharusnya ada wajah yang tidak terdeteksi. Nilai FP merupakan hasil prediksi benar (mendeteksi wajah) dan kenyataannya yang dideteksi bukan lah wajah. Dari nilai-nilai TP, FP dan FN maka peneliti dapat mengetahui precision, sensitivity, dan f-score dari masing-masing model sebagai bahan komparasi.

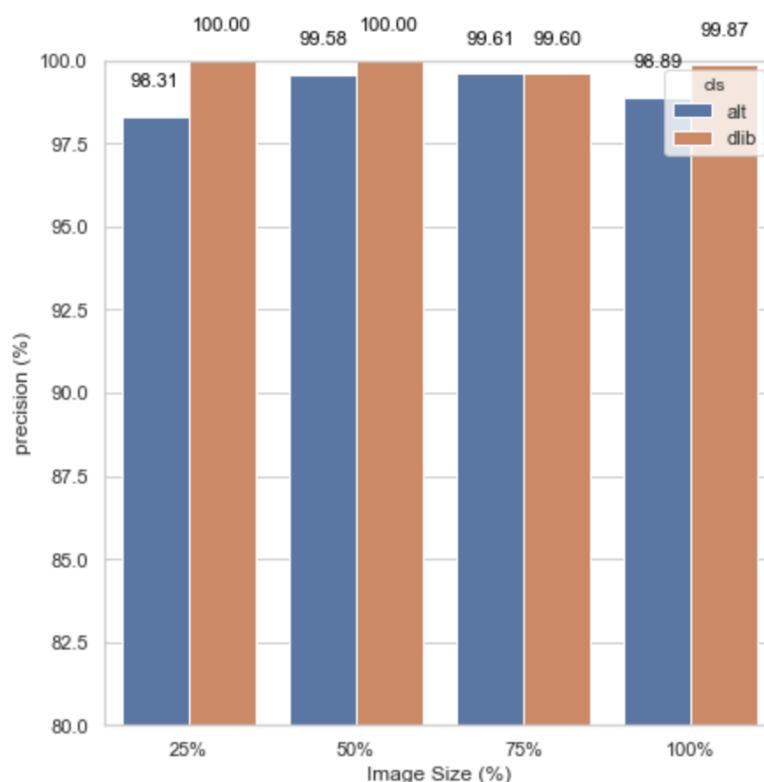
Perbandingan performa nantinya dilihat dengan membandingkan nilai confusion matrix yang dikeluarkan oleh masing-masing model dan divisualisasikan dalam berbagai bentuk grafik. Jupyter notebook digunakan sebagai mediasi visualisasi grafik pada setiap confusion matrix. Precision atau yang juga dikenal sebagai Positive Predictive Value (PPV) yaitu menghitung pada saat diprediksi benar maka seberapa sering bahwa kenyataannya benar. Sensitivity atau Recall biasa disebut sebagai True Positive Rate (TPR) yaitu menghitung pada saat kenyataannya benar, seberapa sering prediksinya

benar. F-Score merupakan rata-rata dari precision dan sensitivity. Semua perhitungan akan dihitung dalam bentuk persentase.

Faktor kecepatan deteksi wajah (dalam waktu dan satuan detik) juga diukur sebagai parameter pembandingan bagi kedua pre-trained classifier yaitu haar cascade dan dlib. Pengukuran kecepatan deteksi wajah akan menggunakan library dari python yaitu timeit. Timeit adalah sebuah library yang dapat menghitung waktu dari kode yang sedang berjalan. Timeit akan diterapkan pada 10 buah foto yang sama dari dataset dengan ukuran 100%, 75%, 50%, dan 25% untuk mengukur kecepatan deteksi wajah dari kedua pre-trained classifier.

IV. HASIL TEMUAN

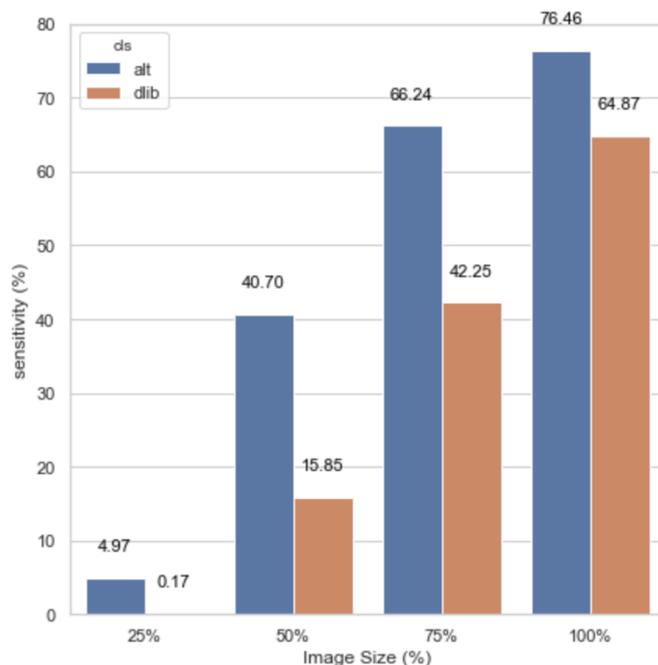
A. Perbandingan Evaluasi Precision atau Positive Predictive Value



Gambar 7 Perbandingan Precision

Dari Gambar 7 dapat dilihat bahwa pada image size 25% dan 50%, precision dari dLib merupakan hasil yang sempurna yaitu 100%. Pada image size 75% kedua classifier memiliki hasil dengan selisih 0.01% namun, pada image size 100% classifier dlib lebih besar dari pada alt yaitu dengan hasil 99.87%. Dari semua image size kedua classifier mengeluarkan hasil yang comparable namun perbedaan yang signifikan terdapat pada image size 25% dan 100%. Dapat disimpulkan bahwa FP yang di deteksi oleh dlib lebih sedikit dibandingkan dengan alt sehingga precision yang dihasilkan oleh dlib lebih besar daripada alt.

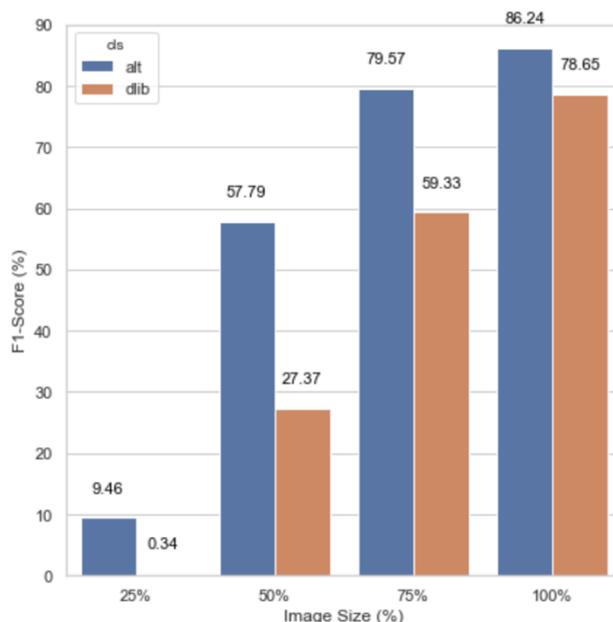
B. Perbandingan Evaluasi Sesitivity atau True Positive Rate



Gambar 8 Perbandingan Sensitivity

Dari Gambar 8 dapat dilihat bahwa pada image size 25% kedua classifier mendapat nilai ter-endah yaitu dibawah 5% dengan dlib paling rendah yaitu 0.17%. Dari semua size, classifier alt memiliki nilai sensitivity yang lebih besar daripada dlib. Dari Gambar 8 dapat disimpulkan bahwa classifier alt mendeteksi TP lebih banyak dibandingkan dengan dlib. jumlah FN yang di deteksi oleh dlib lebih banyak dibandingkan dengan alt sehingga sensitivity dari alt lebih besar dibandingkan dengan dlib.

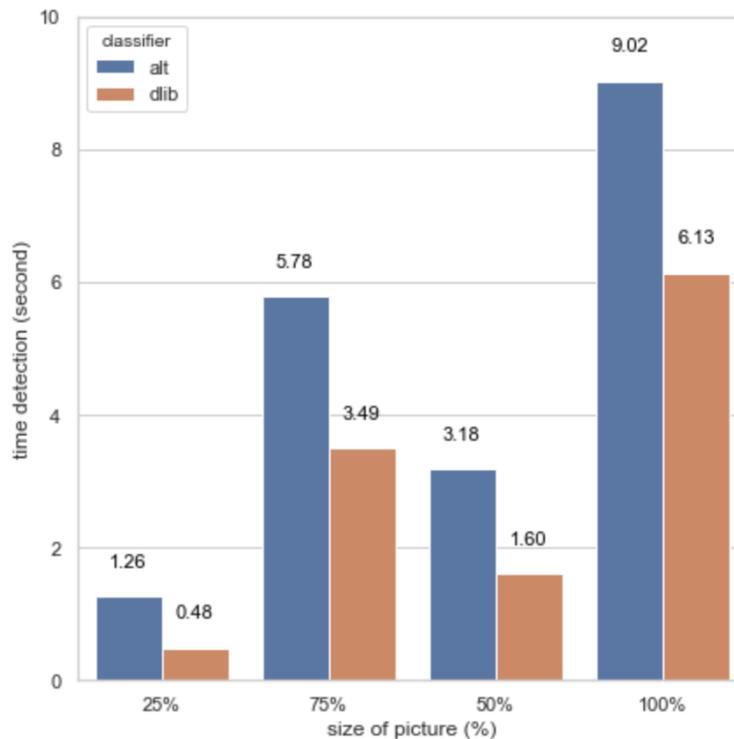
C. Perbandingan F1-Score



Gambar 9 Perbandingan F1-Score

Dari Gambar 9 dapat dilihat bahwa rata-rata dari image size 25% memiliki nilai yang paling rendah dengan classifier dlib paling rendah adalah 0.34%. Dari semua image size classifier alt mempunyai nilai F1-Score lebih besar dibandingkan dengan dlib. Dapat disimpulkan bahwa rata-rata precision dan sensitivity dari alt lebih besar dibandingkan dengan dlib.

D. Perbandingan Waktu Deteksi Wajah



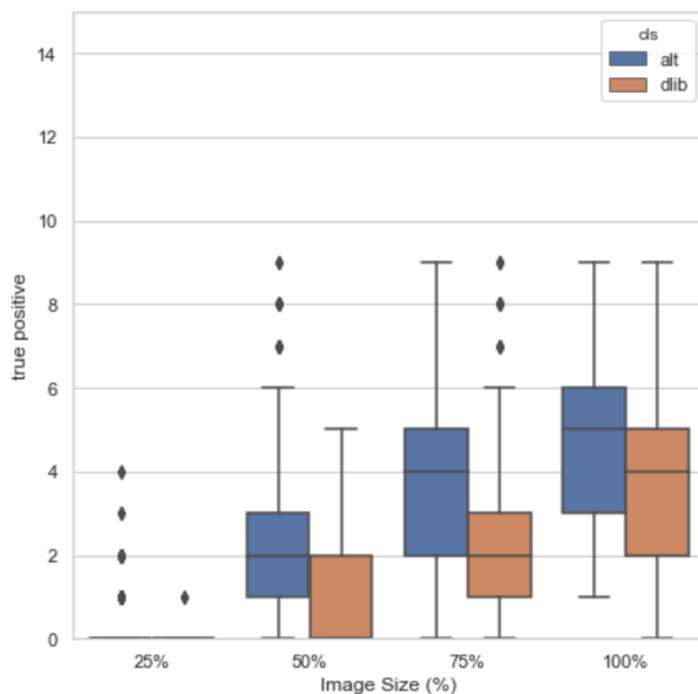
Gambar 10 Waktu Deteksi Wajah

Classifier alt dan *dlib* di-run menggunakan perangkat keras dengan spesifikasi sebagai berikut :

- *Processor* : 2.7GHz dual-core intel core i5 (turbo boost up to 3.1GHz).
- *RAM* : 8GB of 1866MHz LPDDR3.
- *Storage* : 128GB.
- *VGA (Video Graphics Adapter)* : Intel Iris 6100.
- *Sistem Operasi* : macOS High Sierra

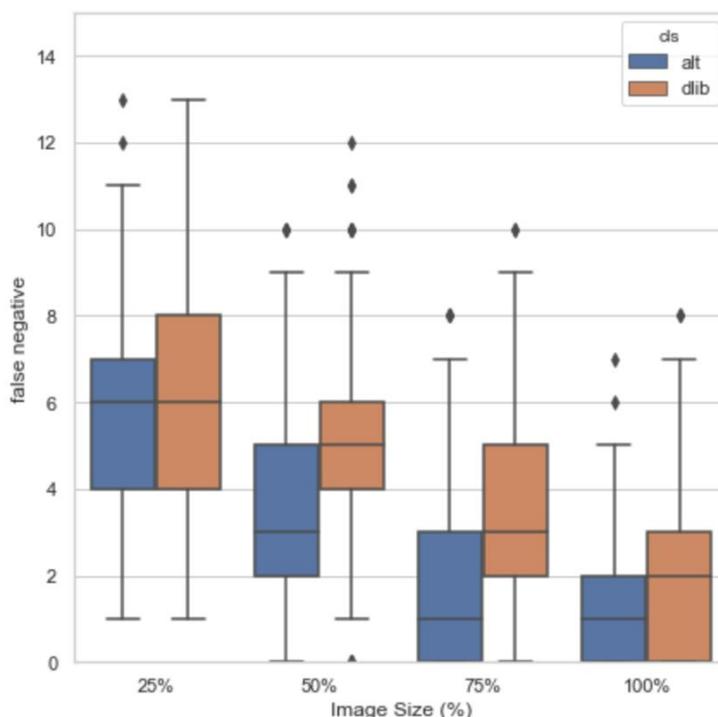
Dengan menggunakan sampel dari dataset dengan image size 25%, 75%, 50% dan 100% (menggunakan 10 gambar yang sama). Perhitungan pada Classifier alt dan dlib dapat disimpulkan bahwa dlib mendeteksi wajah lebih cepat dibandingkan dengan alt pada semua image size.

E. Perbandingan Boxplot Data



Gambar 11 Boxplot True Positive

Gambar 11 adalah boxplot TP dari semua image size pada data. Dilihat dari Gambar 11 penyebaran data pada image size 25% memiliki penyebaran data yang sangat sempit, hal ini kemungkinan karena deteksi TP pada classifier alt dan dlib sangat sedikit. Range pada image size 100% pada classifier alt memiliki nilai 3 sampai 6.



Gambar 12 Boxplot False Negative

Gambar 12 adalah boxplot FN dari semua image size pada data. Dilihat dari Gambar 12 penyebaran data pada image size 25% memiliki penyebaran data yang lebih besar dari image size lain, hal ini kemungkinan karena wajah yang tidak deteksi (FN) pada classifier alt dan dlib sangat banyak. Nilai minimum pada image size 25% yaitu 1 dan nilai maksimum nya adalah 13.

V. KESIMPULAN

Perbandingan yang dibangun melibatkan dua buah *pre-trained classifier* dengan basis yang berbeda namun pada penelitian ini menggunakan *dataset* dari *IBATs* karena penelitian ini berfokus pada penerapan di lingkungan belajar. *Dataset* yang akan dibandingkan diukur dengan ukuran 100%, 75%, 50% dan 25%. Setelah di dapat nilai dari metode evaluasi dan perbandingan ditemukan beberapa poin, yaitu :

- Dalam kondisi wajah yang jaraknya jauh atau orang tersebut duduk di barisan ke-4 atau paling belakang (dalam foto *dataset* wajah tersebut lebih kecil dibandingkan dengan yang duduk di depan) *classifier haar cascade* lebih baik dalam mendeteksi wajah dibandingkan dengan *dlib*.
- Bila ada objek lain yang menyerupai wajah, *classifier alt* lebih sering mendeteksi objek tersebut sebagai wajah (hal ini menimbulkan *FP*).
- Dalam hal kesalahan deteksi wajah (*FP*), *classifier haar cascade (alt)* lebih banyak dibandingkan dengan *dlib*.
- Wajah yang tidak terdeteksi (*FN*) oleh *classifier dlib* lebih banyak dibandingkan dengan *alt*.
- Dalam hal deteksi wajah (*TP*), *classifier alt* lebih banyak dibandingkan dengan *dlib*.
- Dalam durasi deteksi wajah, *classifier dlib* lebih cepat dibandingkan dengan *classifier alt*.
- Bila ingin disimpan dalam *database*, peneliti menyarankan untuk di-*resize* sebanyak 75% dibandingkan dengan ukuran gambar 100% selain mengurangi *storage* pada *database*, hal tersebut juga dapat mempercepat proses deteksi wajah.

Dari poin-poin diatas dapat disimpulkan bahwa *classifier haar cascade* lebih baik dibandingkan dengan *dlib* untuk mendeteksi wajah pada lingkungan pembelajaran Fakultas Teknologi Informasi Universitas Kristen Maranatha.

DAFTAR PUSTAKA

- [1] D. Marčetić, T. Hrkać dan S. Ribarić, "Two-stage cascade model for unconstrained face detection," dalam 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, 2016.
- [2] "dLib," [Online]. Available: https://github.com/ageitgey/face_recognition. [Diakses October 2018].
- [3] "Haar Cascade," [Online]. Available: <https://github.com/opencv/opencv/tree/master/data/haarcascades>. [Diakses October 2018].
- [4] "FaceNet," [Online]. Available: <https://github.com/davidsandberg/facenet>. [Diakses October 2018].
- [5] "Multi-Task Neural Network," [Online]. Available: <https://github.com/ipazc/mtcnn>. [Diakses 12 October 2018].
- [6] Y. Ming-Hsuan, D. J. Kriegman dan N. Ahuja, "Detecting faces in images: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34-58, 2002.
- [7] P. Viola dan M. Jones, "Robust real-time object detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2002.
- [8] "Intel Open Source Computer Vision Library," lore, [Online]. Available: <https://sourceforge.net/projects/opencvlibrary/>. [Diakses November 2018].
- [9] R. Padilla, C. C. Filho dan M. G. F. Costa, "Evaluation of Haar Cascade Classifiers Designed for Face Detection," World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, vol. 6, no. 14, 2012.
- [10] D. E. King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, no. 10, pp. 1755-1758, 2009.
- [11] C. Sagonas, E. Antonaskos, G. Tzimiropoulos, S. Zafeiriou dan M. Pantic, "300 faces In-the-wild challenge: Database and results," dalam Image and Vision Computing (IMAVIS), 2016. "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [12] BMJ, "'Gold standard' is an appropriate term.," BMJ, vol. 305, no. 18, p. 305, 1992.
- [13] D. G. Altman dan J. M. Bland, "Statistic Notes: Diagnostic test 2: predictive values," BMJ, vol. 309, no. 6947, p. 102, 1994.
- [14] D. G. Altman dan J. M. Bland, "Statistics Notes: Diagnostic test 1 : sensitivity and specificity," BMJ, vol. 308, no. 6943, p. 1552, 1994.