Face Recognition Menggunakan Algoritma Haar Cascade Classifier Dan *Convolutional Neural Network*

Bella Hartika[#1], Defri Ahmad[*2]

[#]*Student of Mathematics Departement Universitas Negeri Padang, Indonesia*
[*]*Lecturer of Mathematics Departement Universitas Negeri Padang, Indonesia*
[1]bllhartika@gmail.com
[2]defriahmad88@gmail.com

Abstract — Face recognition is one of the most widely used biometric technologies in the era of the industrial revolution 4.0 such as smart home, security and presence. In the application of face recognition, a method that can detect and perform facial recognition well is needed, so in this study, the authors combined the Haar Cascade Classifier Algorithm and Convolutional Neural Network in facial recognition. Based on the analysis carried out the face detection process using the haar cascade algorithm in data collection and face recognition using the Convolutional Neural Network resulted in an 80% accuracy program with a final loss of 0.50 and the average required for face recognition was 2,935s.

Keywords - Face Recognition, Haar Cascade Classifier, Convolutional Neural Network

Abstrak — *Face recognition* merupakan salah satu teknologi biometrik yang banyak dimanfaatkan pada era revolusi industri 4.0 seperti *smart home*, *security* dan presensi. Dalam penerapan *face recognition* diperlukan metoda yang dapat mendeteksi dan melakukan pengenalan wajah dengan baik, sehingga pada penelitian ini, penulis mengkombinasikan menggunakan Algoritma Haar Cascade Classifier dan *Convolutional Neural Network* dalam melakukan pengenalan wajah. Berdasarkan analisis yang dilakukan proses deteksi wajah menggunakan algoritma haar cascade dalam pengambilan data set dan pengenalan wajah menggunakan *Convolutional Neural Network* menghasilkan akurasi program sebesar 80% dengan final loss 0.50 serta waktu rata - rata yang dibutukan dalam mengenal wajah yaitu sebesar 2,935s.

Kata Kunci − *Face Recognition*, Haar Cascade Classifier, *Convolutional Neural Network*

PENDAHULUAN

Perkembangan teknologi dan ilmu pengetahuan pada era revolusi industri 4.0 berdampak pada terintegrasinya penggunaan *Information Technology* (IT*)* secara global [1]. Salah satunya adalah *Face recognition* merupakan sebuah teknologi pengenalan wajah yang banyak dimanfaatkan pada *smart home*, presensi, dan *security system* [2]. Dalam pengembangannya, sistem pengenalan wajah masih memiliki permasalahan dalam faktor pencahayaan, ekpresi wajah dan perubahan atribut pada wajah serta kondisi citra wajah yang menjadi masukan sistem yang mempengaruhi keakuratan dalam tahap pengenalan [3].

Telah banyak usaha yang dilakukan oleh para peneliti untuk membangun suatu sistem berbasis pengenalan wajah dengan menggunakan berbagai metode yang berbeda-beda meliputi pemilihan ekstraksi ciri dan teknik klasifikasi. Walaupun sudah banyak teknik-teknik pengenalan wajah yang telah dikemukakan dan telah menunjukkan hasil yang signifikan, namun pengenalan wajah yang handal masih sukar didapatkan [4].

Pada penelitian yang dilakukan oleh Zein pada tahun 2018 Algoritma haar cascade digunakan untuk mendeteksi wajah dengan memanfaatkan processing library yang berfungsi sebagai dasar pengolahan dan pendeteksian citra wajah. Sedangkan untuk proses pengenalan wajah metode yang digunakan adalah eigenface berbasis PCA (*Principal Component Analysis*). Dengan 100 kali ujicoba untuk mendeteksi wajah berhasil terdeteksi sebanyak 94 kali benar dan 2 kali salah mengenali dan 4 tidak terdeteksi. Sehingga tingkat keberhasilan akurasi wajah ini sangat tinggi yaitu mencapai 94%.

Viola Jones menggunakan machine learning AdaBoost untuk mengklasifikasi dan mendeteksi sebuah objek of interest dengan menggunakan fitur Haar. Selanjutnya, dilakukan proses pelatihan sistem deteksi menggunakan Cascade Classifier. Kelebihan dari metode Viola Jones ini sistem dapat mendeteksi wajah secara akurat pada berbagai kondisi pencahayaan. Sedangkan kekurangannya adalah deteksi wajah hanya dapat terdeteksi pada kondisi wajah tegak [5].

Penelitian yang sama juga dilakukan oleh dilakukan oleh Santoso pada tahun 2018 yaitu dengan metoda *Convoutional Neural Network* (CNN), Proses pelatihan CNN dengan menggunakan data ukuran 28x28 px dengan 7 lapisan menghasilkan akurasi yang lebih baik

dibandingkan dengan menggunakan 5 lapisan dengan selisih hasil 8,0 % pada saat pengujian. Penggunaan 7 lapisan pada saat pengujian terhadap data testing memperoleh hasil yang baik dengan tingkat akurasi mencapai 98.57%.

Dalam penelitian ini penulis menggunakan library opencv yang berguna untuk pengolahan citra computer vision yang memanfaatkan sebuah *Application Programming Interface* (API) dimana opencv memungkinkan komputer untuk dapat melihat seperti manusia dengan *vision* tersebut sehingga komputer dapat mengambil keputusan, melakukan aksi dan mengenali terhadap suatu objek berdasarkan deteksi wajah. Salah satu Bahasa pemograman yang paling baik dalam menyediakan *library* opencv adalah Bahasa pemograman Python [6].

Dalam proses kerja *face recognition*, diperlukan sebuah algoritma yang mampu mendeteksi objek untuk mempermudah dalam melakukan pengambilan dataset dan metode yang dapat mempelajari data yang diterima sehingga dapat mengenali objek yang diinginkan.

Algoritma Haar Cascade menggunakan metode statistical dalam melakukan pendeteksian wajah. Metode ini menggunakan sample haarlike features. Classifier dalam ini menggunakan gambar berukuran tetap yaitu 24 x 24. Cara kerja haar dalam mendeteksi wajah adalah dengan menggunakan teknik *sliding window* berukuran 24 x 24 pada keseluruhan gambar dan mencari apakah terdapat bagian dari gambar yang berbentuk seperti wajah atau tidak [7].

Convolutional Neural Network merupakan salah satu jenis neural network yang biasanya digunakan dalam pengolahan data image. Konvolusi atau biasa yang disebut dengan *convolution* adalah matriks yang memiiki fungsi melakukan filter pada gambar. *Convolutional Neural Network* memiliki beberapa *layer* yang difungsikan untuk melakukan filter pada setiap prosesnya. Prosesnya disebut dengan proses *training*. Pada proses *training* terdapat 3 tahapan yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*

Dalam proses kerja *face recognition*, diperlukan sebuah algoritma yang mampu mendeteksi objek untuk mempermudah dalam melakukan pengambilan dataset dan metode yang dapat mempelajari data yang diterima sehingga dapat mengenali objek yang diinginkan.

Sehingga dengan ini penulis mengangkat penelitian dengan judul "*FACE RECOGNITION* DENGAN ALGORITMA HAAR CASCADE CLASSIFIER DAN *CONVOLUTIONAL NEURAL NETWORT*"

METODE

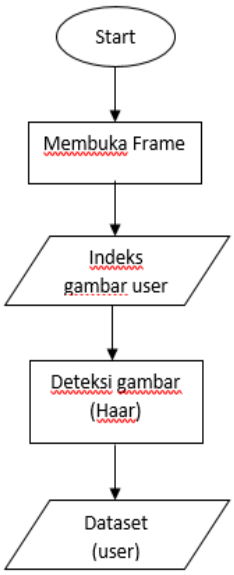
Penelitian ini merupakan penelitian terapan yang diawali dengan studi kepustakaan mengenai konsep sistem pengenalan wajah, *mechine learning* dan selanjutnya mengimplementasikan algoritma Haar Cascade Classifier dan *Convolutional Neural Network* dalam proses *face recognition.*

Data yang digunakan dalam penelitian ini adalah data primer yaitu berupa citra wajah tampak depan dengan berbagai angle pengambilan yang digunakan untuk data training dan data *testing*. Teknik pengumpulan data dilakukan dengan pengambilan citra wajah dengan bantuan kamera. Data citra wajah diambil menggunakan kamera webcam Laptop Lenovo ideaped320 dengan spesifikasi 0.3 MP camera with *single mic*.

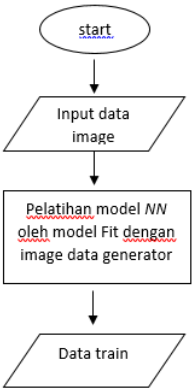
Data yang dianalisis dalam penelitian ini yaitu citra wajah, analisis data dilakukan dengan menentukan tingkat akurasi *face recognition* dengan algoritma haar acsacde classifier dan *Convolutional Neural Network* dengan menghitung berapa waktu komputasi program yang dibutuhkan dalam *face recognition*.

Langkah – langkah analisis data dalam penelitian ini dapat diurutkan sebagai berikut:

1. Melakukan studi pustaka baik dari buku, jurnal, makalah, skripsi dan artikel lainnya mengenai Algoritma haar cascade classifier dan *Convolutional Neural Network*, pengolahan citra dan sistem pengenalan wajah.
2. Mengumpulkan data yang diperlukan dalam proses pengenalan wajah menggunakan kamera Laptop
3. Melakukan *Pre-processing* Citra dengan Algoritma Haar Cascade Classifier.
4. Membuat arsitektur *Convolutional Neural Network* untuk proses pelatihan dan pengujian
5. Melakukan proses Konvolusi pada citra dengan bantuan *library* Keras dan Tensorflow
6. Melakukan proses *Max-Pooling* pada citra untuk mendapatkan
7. Membuat *Fully Connected Layer*



Gambar 1. Flowchart pengambilan dataset



Gambar 2. Flowchart *training* data

Pada tahap ini dilakukan penyalinan Algoritma Haar Cascade Classifier dan *Convolutional Neural Network* dalam format standar ke dalam bahasa pemograman yang digunakan yaitu bahasa python. Terdapat 3 program yang digunakan yaitu sebagai berikut:

1. Program pengambilan dataset wajah tampak *grayscale* yang akan dijadikan data latih dan data testing dengan algoritma Haar
2. Pogram pelatihan untuk mendapatkan data training dengan *library* Keras dan Tensorflow
3. Program *testing* untuk proses pengenalan

Setelah pembuatan program dilakukan simulasi pengujian. Pada tahap ini akan dilakukan simulasi dan pengujian program yang sudah dibuat oleh penulis dan melihat tingkat akurasi program dalam melakukan pengenalan wajah serta dapat menetukan waktu komputasi yang dibutuhkan program dalam melakukan *face recognition*. Menghitung Rata Rata Akurasi Program dengan persamaan matematis yaitu:

$$\bar{A} = \frac{\sum_{n=1}^{10} A_n}{n}$$

Keterangan:
$\bar{A}$ = akurasi rata rata
n = wajah

Waktu komputasi merupakan waktu yang dibutuhakan oleh sistem melakukan suatu proses pengenalan wajah [8]. Adapun secara matematis dapat dituliskan sebagai berikut:

$$\bar{T} = \frac{\sum_{n=1}^{10} T_n}{n}$$

Keterangan:
$\bar{T}$ = rata rata waktu komputasi
n = wajah

HASIL DAN PEMBAHASAN

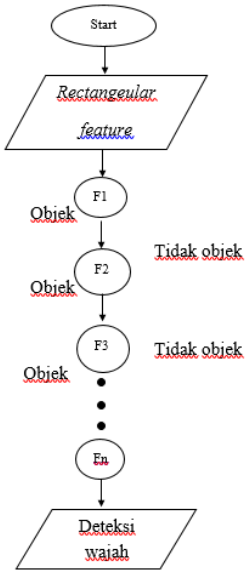
A. Hasil Penelitian

1. Pengumpulan citra

Tahapan pertama yaitu mengumpulkan data citra. Data citra wajah diambil menggunakan kamera webcam Laptop Lenovo ideaped320 dengan spesifikasi 0.3 MP *camera with single mic*. Pengambilan citra dilakukan dengan menggunakan *library* OpenCV dan algoritma haar cascade classifier sehingga menghasilkan data citra grayscale dengan ukuran pixel 140x140. Citra wajah yang diperoleh dibagi menjadi dua kelas yaitu sebagai data training dan data testing. Data ini diambil dari dataset yang telah diperolah sebelumnya kemudian dibagi menjadi dua kelas 80% untuk data t*esting* dan 20% untuk data *training*.

2. *Pre-processing* citra

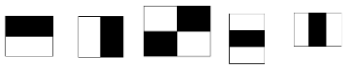
Setelah didapatkan citra, selanjutnya dilakukan *preprocessing* citra berupa *cropping*. *Cropping* citra dilakukan untuk menentukan objek yang diperoses dan dianalisis oleh komputer yaitu citra wajah, sehingga dengan adanya *cropping* pada bagian *background*, komputer tidak perlu menganalisis objek selain citra wajah. Proses *cropping* citra wajah pada penelitian ini menggunakan algoritma haar cascade classfier, sehingga dapat mendeteksi wajah pada gambar dan melakukan *cropping.*



Gamabar 3. *Flowchart* Deteksi Wajah

a. Ekstraksi fitur

1) Ekstraksi fitur dengan haar like fitur



Gambar 4. Fitur yang digunakan

Pada proses ini gambar akan diekstraksi dengan fitur haar, ekstraksi gambar dilakuakan pada basis 24x24 dengan melakuakan *slidding window* dari kiri kekanan dan dari atas kebawah. Nilai fitur dapat dihitung dengan persamaan berikut

$$F\ Haar = \sum F_{putih} - \sum F_{hitam}$$

2) *Integral Image*

Proses *integral image* dilakukan untuk memperoleh nilai fitur dari haar like feature dengan menggunakan tiga operasi aritmatika, yang terdiri dari dua operasi pengurangan dan satu operasi penjumlahan.

Langkah awal yang dilakukan adalah menentuan *integral image* dari sebuah *image* menggunakan Fungsi probabilitas komulatif, yang bertujuan untuk mementukan titik yang mendominasi fitur dalam mendeteksi objek.

$$F(x,y) = P(X \leq x, Y \leq y)$$

$$F(x,y) = \sum_{X \leq x, Y \leq y} P\ (X,Y)$$

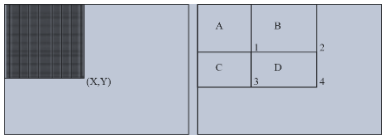
Misalkan nilai piksel sebuah *image* sebagai berikut

2	2	1	0	5	3	4
2	1	0	1	4	4	1
1	0	1	0	2	1	2

sehingga diperoleh nilai integral image berikut ini

2	4	5	5	10	13	17
4	7	7	8	14	21	26
5	8	10	11	22	30	37

Setelah menentukan elemen dari masing –masing *integral image*, selanjutnya menghitung Fitur dari haar like feature, yang diilustrasikan pada Gambar 5



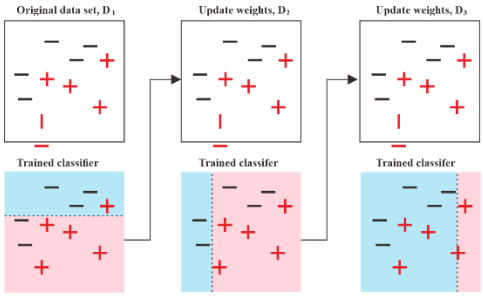
Gambar 5. *Integral Image*

$$D = (A+B+C+D) - (A+B) - (A+C) + A$$

b. Klasifikasi berdasarka fitur

1) Melakukan klasifikasi dengan Adaboost

Algoritma Adaboost learning digunakan untuk meningkatkan kinerja klasifikasi dengan pembelajaran sederhana untuk menggabungkan *classifier* lemah menjadi satu *classifier* kuat. *Classifier* adalah suatu jawaban benar dengan tingkat kebenaran kurang akurat. Sebuah *classifier* lemah dinyatakan pada persamaan berikut



Gambar 6. Adaboost Learning

Penjelasan dari gambar:

a) Pada original data set, dimisalkan terdapat dua kelas yaitu positif dan negatif, langkah awal dari adaboost adalah menetukan *classifier* lemah dengan melihat dimana *error* paling sedikit yang didefenisikan secara matematis berikut ini

$$h_j(x) = \begin{cases} 1, jika\ p_j f_j(x) < p_j \theta_j \\ 0, untuk\ lainnya \end{cases}$$

b) Setelah mendapatkan *classifier* lemah selanjutnya dilakukan penambahan bobot pada tahap kedua, yang bertujuan agar dapat mengklasifikasi secara benar dimana sebelumnya terklasifikasi salah, selanjutnya menentukan klasifikasi yang nilai *error* paling sedikit
c) Tahap ke tiga, menemukan menentukan klasifikasi yang nilai errornya sedikit
d) Tahap ke empat penambahan bobot seperti tahap kedua

e) dan menentukan klasifikasi yang nilai *error* paling sedikit
f) diperoleh 3 *clasifier*, emudian dilakukan kombinasi

Cara mengkombinasikan dari tiga classifier menjadi strong classifier, secara matematis

Hipotesis:

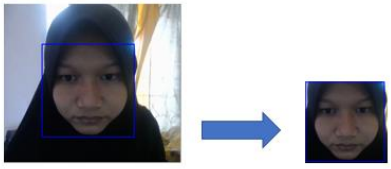
$$H(x) = sign \sum_{t=1}^{T} \alpha_t h_t(x)$$

Langkah langkahnya

1. *training weak classifier*
2. pilih *error* paling kecil
3. beri bobot α_t untuk *classifier* tersebut

$$\alpha_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$$

2) Klasifikasi fitur dengan cascacde classifier, menentukan dibasis mana saja terdapat wajah dan tidak wajah pada gambar



Gambar 7. Hasil deteksi objek dan cropping

Pada penelitian ini menggunkan data *training* dan data *testing*, yang masing-masingnya berjumlah 80% dan 20% dari data set yang telah diambil

3. Pengolahan citra

Pada proses pengolahan citra, data yang diolah adalah hasil segmetasi citra pada data *training* menjadi sebuah matriks, nilai matriks tersebut akan menjadi input pada model yang akan digunakan, dengan menambahkan bobot dan penetapan layer yang akan digunakan dan menambahkan fungsi aktivasi sehingga diperoleh sebuah data pembelajaran yang disebut dengan data training. Nilai matriks diperoleh dengan bantuan *library* matplotlib dan numpy yang digunakan untuk mengkonversi citra graysclae ke bentuk matriks.

a. Konvolusi

.
Operasi konvolusi dapat dituliskan sebagai berikut.

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a)g(x-a)$$

Pada proses konvolusi, $g(x)$ disebut kernel konvolusi atau kernel penapis (*filter*). Kernel $g(x)$ merupakan suatu jendela yang dioperasikan secara bergeser pada pada sinyal masukan $f(x)$. Dalam hal ini, jumlah perkalian kedua fungsi pada setiap titik merupakan hasil konvolusi yang dinyatakan dengan $h(x)$ memberikan output tunggal berupa *feature map*.

Untuk fungsi diskrit dua dimensi berlaku persamaan berikut

$$h(x,y) = \sum_{a=-\infty}^{\infty} \sum_{a=-\infty}^{\infty} f(a)\, w(x-a)$$

Berikut adalah proses konvolusi sebuah image, dimana $f(x)$ merupakan matriks dari sebuah *image* dan $g(x)$ adalah kernel yang digunakan

155	84	95	…	152	230	255
138	104	89	…	132	198	249
111	108	104	…	120	177	233
…						
24	25	26	…	23	19	19
32	30	29	…	20	20	20
33	32	31	…	20	20	20

Fungsi kernel

0	-1	0
1	0	1
0	-1	0

Contoh perhitungan pada proses konvolusi yang sebagai berikut

$$c(1,1) = (155 * 0) + (84 * -1) + (95 * 0) + (138 * 1) + (104 * 0) +$$

$$(89 * 1) + (111 * 0) + (108 * -1) + (104 * 0) = 35$$

Hasil konvolusi diperoleh matriks baru yang disebut dengan *feature layer*, namun tidak dapat dihitung secara manual secara keseluruhannya dikarenakan ukuran matriks yang diperoleh yaitu 226 x 226. Proses konvolusi ini terjadi kepada seluruh data latih yang diinputkan yaitu sebanyak 80% dari data set per masing –masing wajah sehingga total yang melakukan konvolusi adalah 800 citra, yang dilakukan oleh komputer dan dibantu dengan program komputasi.

b. Max-pooling

Pooling Layer merupakan lapisan yang menggunakan fungsi dengan *feature map* sebagai masukan. Pada

penelitian ini ukuran pooling yang digunkaan adalah 2x2 sebanyak 4 kali setelah proses konvolusi. Lapisan pooling yang dimasukkan diantara lapisan konvolusi berturut-turut dalam arsitektur CNN dapat secara progresif mengurangi parameter dan ukuran volume *output* pada *feature map*.

Operasi pada pooling yang digunakan adalah mengambil nilai maksimalnya (max-pooling) dan membuat ukuran output baru menjadi 2x2.

Contoh operasi max-pooling

2	4	6	1
3	1	2	3
2	0	1	2
3	7	0	1

4	6
7	2

Gambar 8. Contoh max-pooling

Hyperparameter:

Filter size (f): 2
Sride (s): 2
Padding: 0

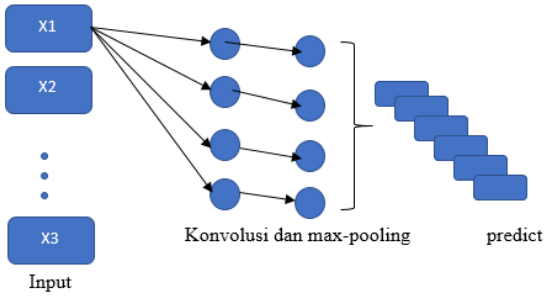
Gambar diatas menunjukan proses dari max-pooling, *output* dari proses pooling adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan pooling akan beroperasi pada setiap irisan kedalaman volume input secara bergantian.

Dari contoh diatas operasi max-pooling dengan menggunakan ukuran *filter* 2x2, masukan pada proses tersebut berukuran 4x4, dari masing masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan dengan membuat ukuran *output* baru menjadi 2x2

4. *Fully connected layer*

Hasil dari proses konvolusi menjadi input *pada fully-connected layer*. *Fully connected layer* adalah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan saraf tiruan biasa. Layer ini tersusun atas beberapa layer yang tiap layer-nya saling terhubung secara penuh (*fully connected*) satu sama lain. Layer ini memliki keluaran yang sama seperti jenis sebelumnya, yaitu berupa vektor yang kemudian ditransformasikan seperti multi-NN dengan beberapa tambahan hidden layer. Hasil keluaran berupa scoring kelas untuk klasifikasi. Model CNN secara matematis, dituliskan

$$y = f\left(w_0 + \sum_{i=1}^{n}(x_i w_i)\right)$$



Gambar arsitektur *fully connected* untuk prediksi

5. Proses komputasi

Sebelum membuat program perlu dilakukan instalisasi Bahasa python dan *library* yang dibutuhkan dalam proses program. Pada penelitian ini dilakukan 3 tahap yang dilakukan yaitu pengambilan dataset, proses *training* dan proses *testing*.

a. Pseudocode program pengambilan data set

1) Membuka kamera dengan library OpenCv
2) *pre-processing*
3) Ekstraksi ciri
4) pengambilan data set menggunakan Cascade Classifier yang *open source*
5) Data citra wajah yang didapatkan akan di simpan di folder database
6) Melakukan pelabelan pada citra
7) Data siap digunakan

b. Pseudocode program pengenalan wajah (testing dan training)

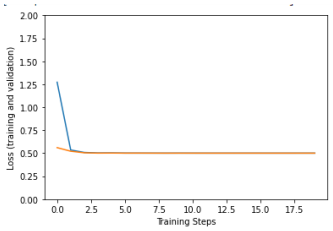
1) Mengimport *libarary*: OpenCv, Numpy, Matplotlib, Keras dan TensorFlow
2) Membuat masing – masing *directory* data *training* dan data *testing*
3) Membuat sub-directory untuk data wajah
4) Membuat image data generator untuk data *training* dan data *testing*
5) Membuat *arsitektur* cnn (cnn yang digunakan 4 convul dan 4 pooling)
6) Memanggil fungsi *compile* pada objek
7) Menentukan *loss function* dan *optimizer*
8) Melatih model (*training* model), Jika program training berhasil, maka dapat dilakukan prediksi model. Jika tidak, maka diulang membuat arsitekstur yang tepat dan memperbaiki data set
9) Membuat prediksi dari model

6. Hasil training model

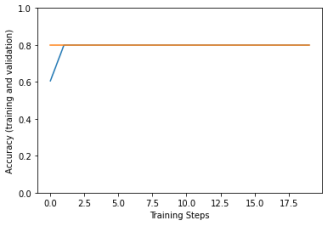
TABEL I. HASIL TRAINING MODEL

Epoch	loss	Accuracy	val_loss	val_ accuracy
1	1.2721	0.6056	0.5604	0.8000
2	5350	0.8000	0.5215	0.8000
3	0.5083	0.8000	0.5044	0.8000
4	0.5024	0.8000	0.5011	0.8000
5	0.5028	0.8000	0.5017	0.8000
6	0.5012	0.8000	0.5009	0.8000
7	0.5008	0.8000	0.5012	0.8000
8	0.5008	0.8000	0.5007	0.8000
9	0.5006	0.8000	0.5006	0.8000
10	0.5008	0.8000	0.5007	0.8000
11	0.5006	0.8000	0.5006	0.8000
12	0.5005	0.8000	0.5006	0.8000
13	0.5005	0.8000	0.5005	0.8000
1,14	0.5005	0.8000	0.5006	0.8000
15	0.5005	0.8000	0.5006	0.8000
16	0.5005	0.8000	0.5005	0.8000
17	0.5005	0.8000	0.5005	0.8000
18	0.5004	0.8000	0.5005	0.8000
19	0.5005	0.8000	0.5006	0.8000
20	0.5005	0.8000	0.5006	0.8000

Dari tabel diatas diperoleh grafik nilai loss pada traning step dan akurasi pada training step, grafik diperoleh dari hasil komputasi dengan *library* matplotlib



Grafik 1. nilai loss pada traning step



Grafik 2. akurasi pada training step

TABEL II. AKURASI DAN FINAL LOSS

Data	Jumlah data	Rata – rata akurasi	Rata – rata Final loss
Training	800	80.00%	0.50
Testing	200	2.1439192e-35	-

TABEL III. WAKTU KOMPUTASI

Uji coba	Waktu Komputasi
1	2.29s
2	3.28s
3	4.32s
4	4,40s
5	3,39s
6	2,95s
7	2,30s
8	2,64
9	1,95s
10	1,83s
Rata-rata	2,935s

SIMPULAN

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Tingkat akurasi data testing yang didapatkan dari Algoritma Haar Cascade Classifier dan Convolutional Neural Network yang terbentuk yaitu sebesar 80 % dengan final loss 0,5 dalam melakukan klasifikasi citra wajah
   .
2. Waktu Rata- rata komputasi yang didapatkan dari Algoritma Haar Cascade Classifier dan Convolutional Neural Network dalam melakukan klasifikasi citra wajah yaitu 2,935s

REFERENSI

[1] Gangopadhyay, I. (2018). Face Detection and Recognition Using Haar Classifier and Lbp Histogram. *International Journal of Advanced Research in Computer Science*, *9*(2), 592–598. https://doi.org/10.26483/ijarcs.v9i2.5815

[2] Septyanto, M. W., Sofyan, H., Jayadianti, H., Simanjuntak, O. S., & Prasetyo, D. B. (2020). APLIKASI PRESENSI PENGENALAN WAJAH DENGAN MENGGUNAKAN ALGORITMA HAAR CASCADE CLASSIFIER. Telematika Jurnal Informatika dan Teknologi Informasi, 16(2), 87-96.

[3] Santoso, A., & Ariyanto, G. (2018). Implementasi deep learning berbasis keras untuk pengenalan wajah. Emitor: Jurnal Teknik Elektro, 18(1), 15-21.

[4] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM computing surveys (CSUR), 35(4), 399-458.

[5] Hardiyanto, D., & Sartika, D. A. (2018). Optimalisasi Metode Deteksi Wajah berbasis Pengolahan Citra untuk Aplikasi Identifikasi Wajah pada Presensi Digital. Setrum: Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer, 7(1), 107-116.

[6] Maryati, R. I. S., & Tryatmojo, B. (2014). Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi Dengan Metode Haar Cascade. Universitas Paradima.

[7] Prathivi, R., & Kurniawati, Y. (2020). SISTEM PRESENSI KELAS MENGGUNAKAN PENGENALAN WAJAH DENGAN METODE HAAR CASCADE CLASSIFIER. Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer, 11(1), 135-142.

[8] Farhan, S. A., Raharjo, J., & Pratiwi, N. K. C. (2019). Identifikasi Wajah Berdasarkan Gender Dan Kelompok Usia Dengan Metode Viola Jones Dan Metode Jaringan Syaraf Tiruan. eProceedings of Engineering, 6(2).