Journal Of Mathematics UNP Vol. 6 No 3 September 2021
P-ISSN/E-ISSN: 2355-1658/2807-3460 12-19

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 a biometric technology that is widely used in the era of the industrial revolution 4.0 such as smart home, security and presence. In the application of face recognition, a method is needed that can perform facial recognition quickly and with a high level of accuracy. This study aims to determine the level of accuracy and computation time of the Haar Cascade Classifier Algorithm and Convolutional Neural Network in Face Recognition with the mechine learning method. Determining the level of accuracy is done by calculating the amount of facial data that can be recognized from the overall face data. Calculating the computational time is done by calculating the time required during the facial recognition process by the computational process. The process of determining the level and time of computing is carried out by the python computing program using the numpy and tensorflow libraries. Based on the analysis carried out by the face detection process using the Haar Cascade Algorithm and Convolutional Neural Network, the program accuracy is 98.84% and the average time needed to recognize faces is 0.05s.

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

Abstrak — *Face recognition* merupakan teknologi biometrik yang banyak dimanfaatkan pada era revolusi industri 4.0 seperti pada *smart home*, *security* dan presensi. Dalam penerapan *face recognition* diperlukan metode yang dapat melakukan pengenalan wajah dengan cepat dan tingkat akurasi yang tinggi. Penelitian ini bertujuan untuk menentukan tingkat akurasi dan waktu komputasi Algoritma Haar Cascade Classifier dan *Convolutional Neural Network* dalam *Face Recognition* dengan metode *mechine learning.* Menentukan tingkat akurasi dilakukan dengan menghitung jumlah data wajah yang dapat dikenali dari data wajah keseluruhan. Menghitung waktu komputasi dilakukan dengan menghitung waktu yang dibutuhkan selama proses pengenalan wajah oleh proses komputasi. Proses menentukan tingkat dan waktu komputasi dilakukan oleh program komputasi python dengan menggunakan librray numpy dan tensorflow. Berdasarkan analisis yang dilakukan proses deteksi wajah menggunakan Algoritma Haar Cascade dan *Convolutional Neural Network* menghasilkan akurasi program sebesar 98.84% dengan serta waktu rata - rata yang dibutukan dalam mengenal wajah yaitu sebesar 0,05s.

Kata Kunci — *Face Recognition,* Haar Cascade Classifier, *Convolutional Neural Network, mechine learning*

PENDAHULUAN

Information Technology merupakan salah satu perkembangan teknologi dan ilmu pengetahuan pada era revolusi 4.0 yang memiliki dampak integrasi global [1].

Salah satunya adalah *Face recognition* merupakan sebuah teknologi pengenalan wajah yang banyak dimanfaatkan pada *smart home*, presensi, dan *security system* [2]. Kendala-kendala seperti pencahayaan, ekspresi dan atribut serta kondisi citra *input* yang mempengaruhi akurasi dalam tahap sistem pengenalan wajah [3]. Telah banyak peneliti menggunakan metode yang berbeda-beda dalam sistem pengenalan wajah ini. Salah satunya teknik ekstraksi dan klasifikasi. Namun belum dapat menunjukkan hasil yang signifikan [4].

Zein pada tahun 2018 melakukan penelitian menggunakan Algoritma haar cascade untuk mendeteksi wajah dengan memanfaatkan *library image processing* . Sedangkan dalam pengenalan wajah metode yang digunakan adalah eigenface berbasis PCA (*Principal Component Analysis*). Dengan pengujian sebanyak 100 kali, dengan hasil terdeteksi sebanyak 94 kali benar dan 2 kali salah mengenali dan 4 tidak terdeteksi. Sehingga tingkat akurasi wajah ini sangat tinggi yaitu sebesar 94%.

Viola dan Jones dalam mengklasifikasi dan mendeteksi sebuah objek of interest dengan menggunakan fitur Haar menggunakan metode *machine*

learning AdaBoost. Selanjutnya, dilakukan *training* sistem deteksi menggunakan Cascade Classifier. Kelebihan dari metode ini adalah sistem dapat mendeteksi wajah dengan akurat diberbagai 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 *training* CNN pada data ukuran 28x28 px dengan 7 *layer* menghasilkan akurasi yang lebih baik dibandingkan dengan menggunakan 5 *layer* dengan selisih hasil 8,0 % pada saat pengujian. Akurasi penggunaan 7 lapisan pada saat pengujian terhadap data testing 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 subset dari neural network yang digunakan untuk *detection image*. Pada CNN terdapat proses Konvolusi yaitu operasi dua deret yang menghasilkan deret barur. *Convolutional Neural Network* memiliki proses *training* yang memiliki beberapa *layer* yang difungsikan untuk melakukan filter. Proses ini terdiri atas 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 citra wajah dan metode yang dapat mempelajari data citra wajah yang diterima sehingga dapat mengenali objek wajah yang diinginkan. Dalam penelitian ini proses pengenalan wajah menggunakan algoritma haar cascade *classifier* dan *convolutional neural network*.

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*.

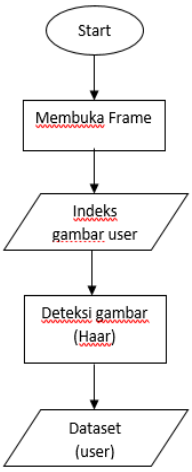
Data yang dianalisis dalam penelitian ini yaitu citra wajah, analisis data dilakukan dengan menentukan tingkat akurasi *face recognition* dengan algoritma haar acsacde classifier dan *Convolutional Neural Network* dengan menghitung berapa waktu komputasi program yang dibutuhkan dalam *face recognition*.

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

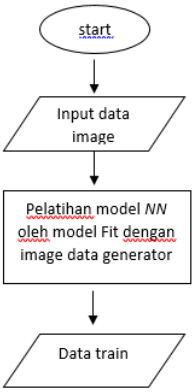
1. Melakukan studi pustaka baik dari buku, jurnal, makalah, skripsi dan artikel lainnya mengenai Algoritma haar cascade classifier dan *Convolutional Neural Network*, pengolahan citra dan sistem pengenalan wajah.
2. Mengumpulkan data yang diperlukan dalam proses pengenalan wajah menggunakan kamera Laptop
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*
8. Melakukan proses Konvolusi pada citra dengan bantuan library Keras dan Tensorflow
9. Melakukan proses Max Pooling pada citra untuk mendapatkan
10. Membuat program komputasi dengan bahasa python
11. Melakukan proses *tarining*
12. Menentukan proses *testing*
13. Mengitung akurasi program komputasi dalam proses *face recogniton*
14. Menghitung waktu komputasi progam

Proses komputasi pada penelitian ini dilakukan pada google colab dan IDLE python dengan memanfaatkan *library* numpy, matplotlib, keras dan tensorflow. pada pengolahan citra dilakukan proses konvolusi sebanyak

tiga kali, hasil yang diproses selanjutnya dilakukan maxpooling selanjutnya *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 sebagai berikut

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

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

Waktu komputasi merupakan waktu yang dibutuhkan 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. *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*.

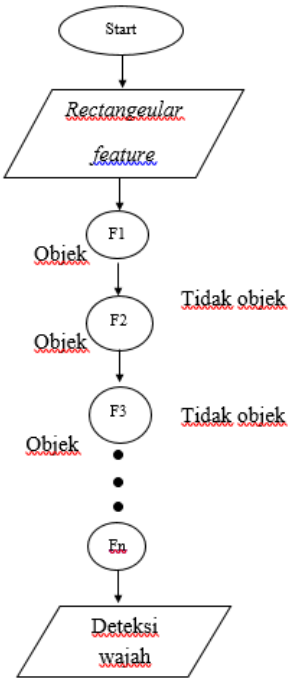
B. *Pre-Processing Citra*

Selanjutnya melakukan *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*.

Cara kerja haar dalam mendeteksi wajah adalah dengan menggunakan teknik *sliding window* dengan basis yang berukuran 24 x 24 pada keseluruhan gambar dan mencari apakah terdapat bagian dari gambar yang berbentuk seperti wajah atau tidak. Kemudian dilakukan *scaling* sehingga dapat mendeteksi adanya wajah yang berukuran lebih besar ataupun lebih kecil dari gambar yang terdapat pada *classifier*.

Pada *processing* citra terdapat pengklasifikasian objek wajah dan tidak wajah dengan algoritma haar pada proses cascade *classifier* Pada peneliian ini digunakan 3 fitur untuk mendeteksi objek pada haar yaitu, *edge*

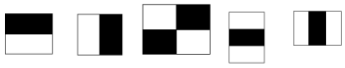
features, *line features*, tipe *four-rectangle feature* dan dibantu dengan library OpenCV dalam deteksi objek.



Gamabar 3. *Flowchart* Deteksi Wajah

1. Ekstraksi fitur

Proses ekstraksi fitur dilakukan untuk mendapatkan nilai fitur dan mennetukan dimana saja terdeteksi objek wajah pada 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

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

Untuk mempermudah dalam menghitung nilai fitur dilakukan proses *integral image*, 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.

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

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

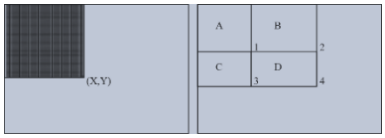
Misalkan nilai piksel sebuah *image* sebagai berikut

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

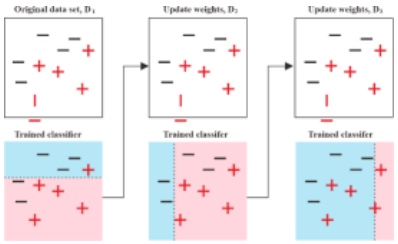


Gambar 5. *Integral Image*

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

2. Klasifikasi berdasarkan fitur

Klasifikasi fitur dilakukan dengan Adaboost learning yang bertujuan untuk melatih *classifier* lemah menjadi clasifier kuat yang dibantu dengan penambahan bobot pada setiap iterasi.



Gambar 6. Adaboost Learning

Penjelasan dari gambar 6:

a) Pada original dataset, 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:

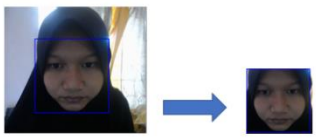
$$H(x) = sign \sum_{t=1}^{T} \alpha_t h_t(x)$$

Langkah langkahnya

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

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

Selanjutnya melakukan klasifikasi fitur dengan cascacde classifier, menentukan dibasis mana saja terdapat wajah dan tidak wajah yang dilakukan dengan *teknik slidding window*



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

C. *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.

1. Konvolusi

Konvolusi merupakan sebuah operasi yang menggunakan dua fungsi utnutk menghasilkan sebuah fungsu baru. Operasi konvolusi dapat dituliskan sebagai berikut.

$$h(x) = f(x) * g(x)$$

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

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

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.

2. Max-pooling

Pooling Layer merupakan lapisan setelah proses konvolusi dimana input *pooling layer* adalah hasil dari proses konvolusi. Pada penelitian ini ukuran pooling yang digunkaan adalah 2x2 sebanyak 4 kali setelah proses konvolusi. Untuk mengurangi dimensi dan parameter feature map dilakukan dengan memasukkan lapisan pooling diantara lapisan konvolusi berturut-turut dalam arsitektur CNN. Operasi pooling yang digunakan pada penelitian in adalah max-pooling dengan mengambil nilai maksimalnya

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

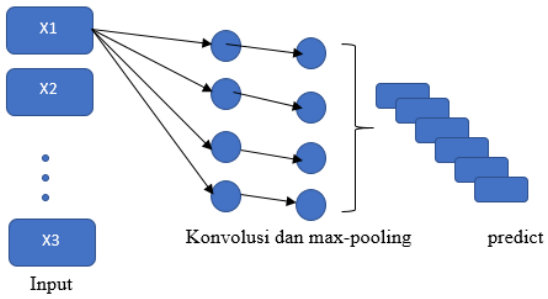
Gambar diatas menunjukan proses dari max-pooling, *output* dari proses pooling adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal.

Operasi max-pooling yang dilakukan adalah 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

D. *Fully connected layer*

Fully-connected layer adalah hasil dari proses konvolusi menjadi input. *Fully connected layer* adalah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron dilapisan 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 sebagai berikut:

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



Gambar 9. Arsitektur *fully connected* untuk prediksi

E. *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*.

1. Pseudocode program pengambilan data set
   a. Membuka kamera dengan library OpenCv
   b. pre-processing
   c. Ekstraksi ciri
   d. pengambilan data set menggunakan Cascade Classifier yang open source
   e. Data citra wajah yang didapatkan akan di simpan di folder database
   f. Melakukan pelabelan pada citra
   g. Data siap digunakan

2. Pseudocode program pengenalan wajah (testing dan training)
   a. Mengimport *libarary*: OpenCv, Numpy, Matplotlib, Keras dan TensorFlow
   b. Membuat masing – masing *directory* data training dan data testing
   c. Membuat *sub-directory* untuk data wajah
   d. Membuat image data generator untuk data training dan data testing
   e. Membuat arsitektur cnn
   f. Memanggil fungsi compile pada objek
   g. Menentukan *loss function* dan *optimizer*
   h. 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
   i. Melakukan testing pada model dengan menginputkan dataset testing
   j. Melakukan uji coba dengan input yang bukan wajah

F. *Hasil Training Model*

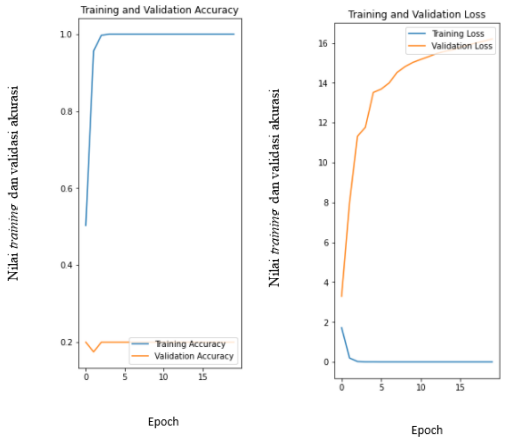
Hasil training model diperoleh dengan proses komputasi melalui menggunakan bahasa python dan *library* tensorflow dan keras.

TABEL I

HASIL TRAINING MODEL

Epoch	*loss*	*accuracy*	*val_loss*	*val_ accuracy*
1	1.7199	0.5031	3.2873	0.2000
2	0.1947	0.9563	7.9904	0.1750
3	0.0210	0.9969	11.3161	0.2000
4	0.0030	1.0000	11.7703	0.2000
5	0.0036	1.0000	13.5160	0.2000
6	3.9992e-04	1.0000	13.6854	0.2000
7	2.9646e-04	1.0000	13.9937	0.2000
8	1.6635e-04	1.0000	14.5214	0.2000
9	9.5824e-05	1.0000	14.8098	0.2000
10	7.7488e-05	1.0000	15.0243	0.2000
11	6.5101e-05	1.0000	15.1724	0.2000
12	5.7503e-05	1.0000	15.3155	0.2000
13	5.1511e-05	1.0000	15.4726	0.2000
14	4.6499e-05	1.0000	15.5728	0.2000
15	4.2400e-05	1.0000	15.6523	0.2000
16	3.8863e-05	1.0000	15.7986	0.2000
17	3.6035e-05	1.0000	15.9071	0.2000
18	3.2804e-05	1.0000	16.0001	0.2000
19	3.0227e-05	1.0000	16.0937	0.2000
20	2.8035e-05	1.0000	16.2000	0.2000

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



Gambar 10. Grafik Nilai *Loss* Pada Proses *Training*

Dari Gambar 10 dinterpretasikan bahwasannya nilai akurasi pada proses *trining* dan validasi. Untuk *training* nilainya garfik menuju 1.0 yang berarti akurasi dalam mengenal wajah baik sehingga dapat melakukan proses pelatihan dan nilai validasi nya 0.20 bearti validasi dalam pelatihan menuju 0.2 setiap pelatihan citra wajah. Untuk training loss menuju 0 yang berarti nilai *loss* pada saat *training* dalam komputasi akan semakin kecil berdasarkan perulangan yang dilakukan

TABEL II

HASIL AKURASI DAN WAKTU KOMPUTASI

No	Wajah	Akurasi	Prediksi	Waktu
1	Tomi	99.99%	Terdefinisi	0.5s
2	Bella	94.52%	Terdefinisi	0.5s
3	Santi	100.00%	Terdefinisi	0.4s
4	Afdal	100.00%	Terdefinisi	0.5s
5	Randa	100.00%	Terdefinisi	0.4s
6	Chindi	100.00%	Terdefinisi	0.5s
7	Deni	100.00%	Terdefinisi	0.5s
8	Amel	98.37%	Terdefinisi	0.5s
9	Putri	96.88%	Terdefinisi	0.5s
10	Habib	99.68%	Terdefinisi	0.4s

TABEL III

RATA-RATA AKURASI DAN WAKTU KOMPUTASI

Data	Jumlah data	Rata – rata akurasi	Rata – rata waktu komputasi
Pengenalan Wajah	10	98.94%	0.5s

Pada Tabel II dan Tabel III diperoleh nilai dan rata – rata akurasi dan waktu komputasi pengenalan wajah menggunakan haar cascade dan *convolutional neural network*, proses haar dilakukan untuk deteksi objek dan cnn berguna untuk melatih data dan melakukan klasifikasi citra dengan proses komputasi.

SIMPULAN

Didapatkan simpulan berdasarkan analisis yang telah dilakukan sebagai berikut:

1. Tingkat akurasi data testing yang didapatkan dari Algoritma Haar Cascade Classifier dan Convolutional Neural Network yang terbentuk yaitu sebesar 98.94%.
2. Waktu Rata- rata komputasi yang didapatkan dari Algoritma Haar Cascade Classifier dan Convolutional Neural Network dalam melakukan klasifikasi citra wajah yaitu 0.05s.

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).