
Pengenalan Ekspresi Wajah Dengan Metode *Viola Jones* dan *Convolutional Neural Network*

Reza Rizki Reynaldo¹, Irfan Maliki^{2*}

¹Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

*email: irfan.maliki@email.unikom.ac.id

(Naskah masuk: tgl. 25 Nopember 2020; diterima untuk diterbitkan: 04 Maret 2021)

ABSTRAK – Saat ini penggunaan kecerdasan buatan berkembang dengan pesat, diantaranya dapat dimanfaatkan untuk mengenali ekspresi wajah manusia. Komputer memainkan peranan penting dalam interaksi komputer dan manusia dengan mengenali ekspresi wajah manusia. Pengenalan ekspresi wajah manusia ini telah banyak digunakan ke dalam robot, kamera pengawas keamanan, game interaktif, serta dapat digunakan untuk memainkan player musik didasarkan pada ekspresi wajah. Pengenalan ekspresi wajah manusia secara otomatis memiliki tingkat kompleksitas yang tinggi dan sulit. Tujuan yang ingin dicapai dari penelitian ini adalah untuk mengetahui dan meningkatkan akurasi pengenalan ekspresi wajah dengan menggunakan metode *Viola Jones* dan *Convolutional Neural Network*. Metode yang dilakukan dalam penelitian ini terdiri dari preprocessing, segmentasi dengan menggunakan *Viola Jones* dan klasifikasi dengan metode *Convolutional Neural Network*. Dataset yang digunakan sebanyak 2205 data ekspresi wajah yang terdiri dari bahagia, marah, muak, sedih, takut, terkejut dan ekspresi normal. Sumber data diperoleh dengan cara menangkap ekspresi wajah secara langsung dari 42 orang responden. Berdasarkan hasil pengujian yang telah dilakukan dengan menggunakan confusion matrix diperoleh tingkat akurasi sebesar 96,14% dengan parameter epoch 10 dan learning rate 0,001. Dari hasil pengujian ini menunjukkan bahwa metode yang diusulkan memiliki akurasi yang lebih baik untuk mengenali ekspresi wajah secara otomatis.

Kata Kunci - pengenalan ekspresi wajah, *viola jones*, *convolutional neural network*, segmentasi, klasifikasi.

Facial Expressions Recognition using *Viola Jones* and *Convolutional Neural Network* Method

ABSTRACT – Currently the use of artificial intelligence is growing rapidly, including it can be used to recognize human facial expressions. Computers play an important role in computer and human interaction by recognizing human facial expressions. Human facial expression recognition has been widely used in robots, security surveillance cameras, interactive games, and can be used to play music players based on facial expressions. Automatic recognition of human facial expressions has a high level of complexity and difficulty. The purpose of this research is to determine and improve the accuracy of facial expression recognition using the *Viola Jones* and the *Convolutional Neural Network* method. The method used in this study consisted of preprocessing, segmentation using *Viola Jones* and classification using *Convolutional Neural Network* method. The data set used was 2205 facial expressions consisting of happiness, anger, disgust, sadness, fear, surprise and normal expressions. Sources of data obtained by capturing facial expressions directly from 42 respondents. Based on the results of tests that have been done using the confusion matrix, an accuracy rate of 96.14% is obtained with epoch 10 parameters and a learning rate of 0.001. The results of this test show that the proposed method has better accuracy for recognizing facial expressions automatically.

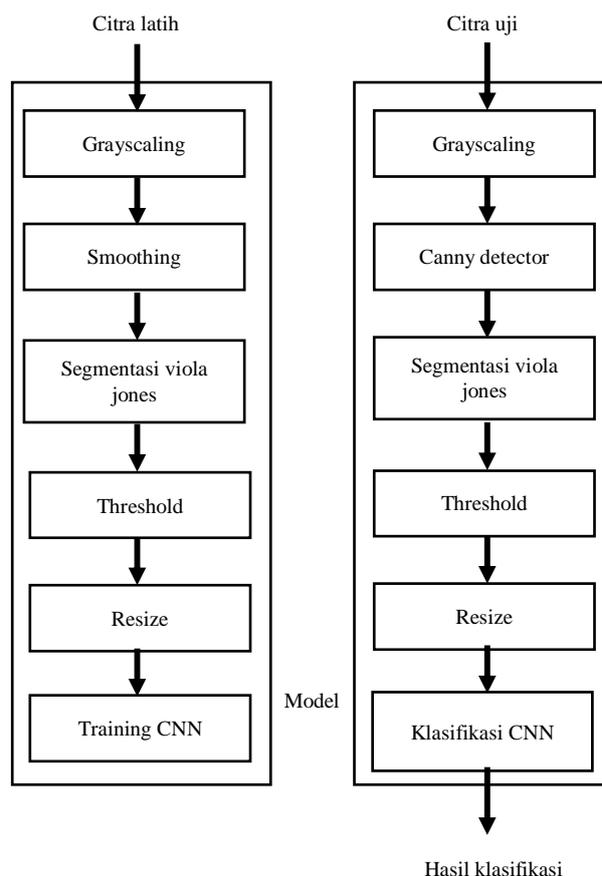
Keywords – facial expression recognition, *viola jones*, *convolutional neural network*, segmentation, classification.

1. PENDAHULUAN

Ekspresi wajah manusia merupakan bentuk respon alami manusia yang menggambarkan perasaan yang dirasakan oleh seseorang saat berinteraksi pada suatu hal tertentu. Ekspresi wajah mengacu pada komunikasi nonverbal yang sangat kuat yang digunakan untuk berkomunikasi. Ekspresi seperti marah, muak, takut, bahagia, sedih, netral dan terkejut merupakan Tujuh ekspresi wajah dasar [1]. Tujuh ungkapan ini secara luas dikategorikan menjadi emosi positif dan negatif. Ekspresi terkejut, bahagia dan terkejut termasuk ke dalam ekspresi positif dan takut, sedih, marah dan muak termasuk ke dalam ekspresi negatif [2]. Komputer memainkan peranan penting dalam interaksi komputer dan manusia dengan mengenali ekspresi wajah manusia. Pengenalan ekspresi manusia ini telah banyak digunakan ke dalam robot, kamera pengawas keamanan dan game interaktif, serta dapat digunakan untuk memainkan player musik didasarkan pada ekspresi wajah. Pengenalan ekspresi wajah secara otomatis memiliki tugas yang kompleks karena wajah sangat variatif dari satu individu ke individu lainnya [3]. Salah satu metode yang dapat digunakan untuk mengenali ekspresi wajah, yaitu dengan *Convolutional Neural Network* (CNN). Metode ini memiliki akurasi yang cukup baik seperti dalam penelitian Levi dkk [4] pada klasifikasi usia dan jenis kelamin menghasilkan akurasi sebesar 86,80%. Penelitian Jaswal [5] dengan hasil akurasi diatas 95%. Penelitian Zhou [6] menghasilkan akurasi lebih dari 90%. Penelitian Antipov [7] menghasilkan akurasi diatas 90%. Namun metode CNN mengalami penurunan akurasi pada pengenalan ekspresi wajah seperti yang dilakukan oleh Zheng [8] dengan hasil tingkat akurasi sebesar 61,19%. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengetahui dan meningkatkan akurasi pengenalan ekspresi wajah dengan memperbaiki tahap preprocessing segmentasi dengan metode Viola Jones dan klasifikasi dengan metode CNN.

2. METODE

Metode yang dilakukan pada penelitian ini yaitu meliputi tahap pengumpulan data, tahap *preprocessing*, dan tahap klasifikasi. Data yang telah didapat akan dilakukan *preprocessing* dimana data gambar akan dirubah menjadi citra keabuan (*grayscale*), kemudian dilakukan proses penghalusan citra (*smoothing*), lalu dilakukan proses segmentasi dengan metode *Viola Jones*, kemudian dilanjutkan pada tahap perubahan terhadap ukuran gambar (*resize*). Selanjutnya melakukan proses klasifikasi menggunakan metode CNN, untuk lebih jelasnya dapat dilihat pada Gambar 1.



Gambar 1. Metode pengenalan ekspresi wajah

2.1. Grayscale

Pada tahapan ini citra masukan berwarna (RGB) dirubah menjadi citra berskala keabuan (*grayscale*). Proses ini menggunakan persamaan 1. [9]

$$I = 0.2989 \times R + 0.5870 \times G + 0.1141 \times B \quad (1)$$

2.2 Smoothing

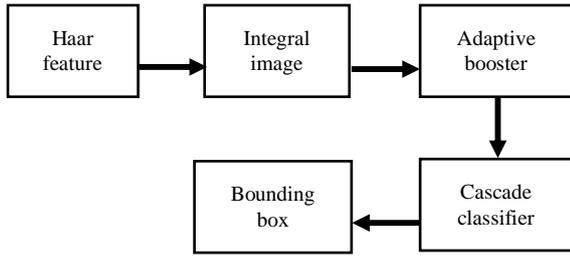
Setelah citra masukan diubah menjadi citra keabuan dilakukan proses *smoothing* atau penghalusan citra untuk memperbaiki kualitas citra [10]. Metode yang digunakan untuk penghalusan ini adalah metode *Gaussian smoothing*. Metode ini menggunakan fungsi *gaussian* yang beroperasi dengan cara mengkonvolusikan citra dengan kernel *gaussian* [9]. Untuk merubah citra menjadi lebih halus dapat dilakukan menggunakan persamaan 2.

$$h(x, y) = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

2.3 Segmentasi Viola Jones

Selanjutnya dilakukan proses segmentasi yang bertujuan untuk menentukan area wajah pada citra. Citra masukan akan diproses dengan cara memotong gambar yang sudah terfokuskan pada bagian wajah. Proses segmentasi ini memanfaatkan metode *haar-like feature* dari algoritma *Viola Jones* untuk mendeteksi

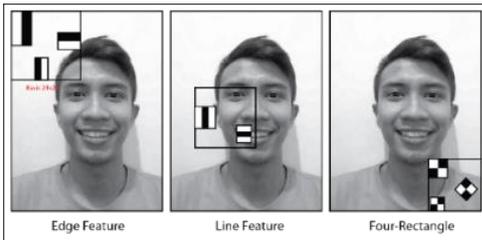
objek wajah pada citra masukan. Adapun tahapannya dapat dijelaskan pada gambar 2.



Gambar 2. Tahapan Viola Jones

A. Haar Like Feature

Tahap awal dalam proses segmentasi *Viola Jones* adalah *haar like feature*. Teknik ini berfungsi untuk mendeteksi wajah dengan cara mengotak-kotakan setiap daerah pada citra dari mulai ujung kiri atas hingga ujung kanan bawah. Proses ini dilakukan untuk mendeteksi apakah ada fitur wajah pada area tersebut. Ilustrasi proses ini dapat pada Gambar 3.



Gambar 3. Ilustrasi Haar Like Feature

B. Integral Image

Proses *integral image* digunakan untuk mempercepat proses pendeteksian objek. Nilai-nilai piksel hasil dari proses *haar like feature* akan dihitung dengan persamaan 3. [11]

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3)$$

Setelah mendapatkan hasil dari proses perhitungan *integral image*, dilakukan penghitungan area tertentu. Untuk menghitung wilayah tertentu dapat dilakukan dengan menggunakan persamaan 4. [11]

$$D = L1 + L4 - (L2 + L3) \quad (4)$$

C. Adaptive Booster

Adaptive Booster merupakan teknik yang digunakan untuk mengkombinasikan banyak *classifier* lemah untuk membentuk suatu gabungan *classifier* yang lebih baik. Proses dari metode ini akan menghasilkan sebuah *classifier* yang kuat dari *classifier* dasar. *Adaptive Boosting* mengkombinasikan *performance* banyak *weak classifier* untuk

menghasilkan *strong classifier*. *Weak classifier* dalam hal ini adalah nilai dari *haar-like feature* [9]. Langkah pertama yang harus dilakukan dalam tahapan ini adalah dengan menghitung bobot awal kemudian menghitung nilai fitur dari sebuah *weak classifier*. Adapun persamaan yang digunakan pada tahapan ini adalah persamaan 5, 6 dan 7. [11]

Bobot awal

$$w_{j1yi} = \frac{1}{2m}, w_{j1yi} = \frac{1}{2i} \quad (5)$$

Untuk citra positif

$$\epsilon_t = \left(\sum_t^T w_{t,i} \right) |h_t(x) - (y_i)| \quad (6)$$

Untuk citra negatif

$$\epsilon_j = \left(\sum_j^J w_{t,i} \right) |h_j(x) - (y_i)| \quad (7)$$

Di mana:

w = *week classifier*

m = jumlah citra positif.

l = jumlah citra negatif.

t = indeks iterasi dari citra positif.

j = indeks iterasi dari citra negatif.

$h(x)$ = nilai fitur citra positif.

$h(x)$ = nilai fitur citra negatif.

Setelah mendapatkan nilai fitur dari sebuah *weak classifier*. Maka apabila nilai fitur kurang dari 0 maka citra tersebut dianggap tidak memiliki objek [11].

$$H(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{bukan} \end{cases} \quad (8)$$

Di mana:

$$\alpha_j = \log \frac{1}{\beta_j}, \alpha_t = \log \frac{1}{\beta_t} \quad (9)$$

Kondisi:

Jika posisi $H(x) =$ Ketentuan 1 maka citra tersebut merupakan objek

Jika posisi $H(x) =$ Ketentuan 0 maka citra tersebut merupakan bukan objek

Keterangan:

$H(x)$ = Strong Classifier atau klasifikasi yang menyatakan objek atau bukan

α_j = Tingkat pembelajaran citra positif

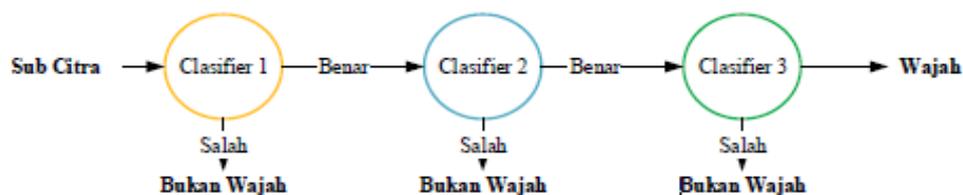
α_t = Tingkat pembelajaran citra negatif

β_t = Nilai bobot setelah *error rate* pada citra +

β_j = Nilai bobot setelah *error rate* pada citra -

H_t = *week* atau *basic classifiers* (awal dari klasifikasi) citra +

H_j = *week* atau *basic classifiers* (awal dari klasifikasi) citra -



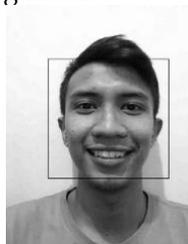
Gambar 4. Cascade Classifier [11]

D. Cascade Classifier

Pada proses *cascade classifier* yang merupakan metode untuk mengkombinasikan *classifier* yang kompleks dalam sebuah struktur yang bertingkat dan dapat meningkatkan kecepatan pendeteksian sebuah objek pada citra yang memfokuskan pada daerah citra yang berpeluang saja, dan gambar 4 merupakan proses dari *cascade classifier*.

E. Bounding Box

Proses ini merupakan proses yang memanfaatkan *Bounding Box* untuk menandai objek wajah yang sudah terdeteksi dengan tanda area yang berbentuk *box*. Proses ini digunakan untuk mempersempit area penelitian yang lebih terfokus pada bagian wajah, yang akan diterapkan pada data citra latih dan citra uji. Proses ini merupakan proses lanjutan dari proses yang sebelumnya. Adapun hasil dari *bounding box* ditampilkan pada gambar 5.



Gambar 5. Bounding Box

Selanjutnya area yang sudah ditandai dilakukan pemotongan pada area tersebut, hal ini diperlukan untuk mempermudah dalam menemukan kontur wajah pada tahap selanjutnya.



Gambar 6. Citra yang Telah Terpotong

Setelah dilakukan pemotongan pada area wajah seperti pada gambar 6, ukuran citra berubah menjadi 509x509 piksel dari citra masukan yang berukuran 778x1037 piksel.

2.4. Thresholding

Selanjutnya dilakukan proses *thresholding* untuk mengubah nilai citra menjadi nilai 0 dan 255 [9]. Nilai *threshold* yang digunakan adalah 127. Adapun ketentuannya dapat dilihat pada persamaan 10.

$$(x, y) = \begin{cases} 0, & \text{jika grayscale}(x, y) < 127 \\ 255, & \text{selain itu} \end{cases} \quad (10)$$

2.5 Resize

Pada tahapan ini akan dilakukan perubahan ukuran citra masukan. Proses ini diperlukan agar pada saat tahap pelatihan dan pengujian citra memiliki ukuran yang sama, selain itu juga untuk mempercepat proses komputasi. Dari hasil percobaan, citra akan diubah ukurannya menjadi 220 x220. Adapun persamaan yang digunakan untuk merubah ukuran citra dapat dilihat pada persamaan 11 dan 12.

$$x = \frac{pb * pp}{pa} \quad (11)$$

x menunjukkan posisi x baru, pb untuk menentukan ukuran panjang dari matriks baru, pp menunjukkan nilai pixel lama, dan pa menunjukkan ukuran panjang matriks lama.

$$y = \frac{lb * pp}{la} \quad (12)$$

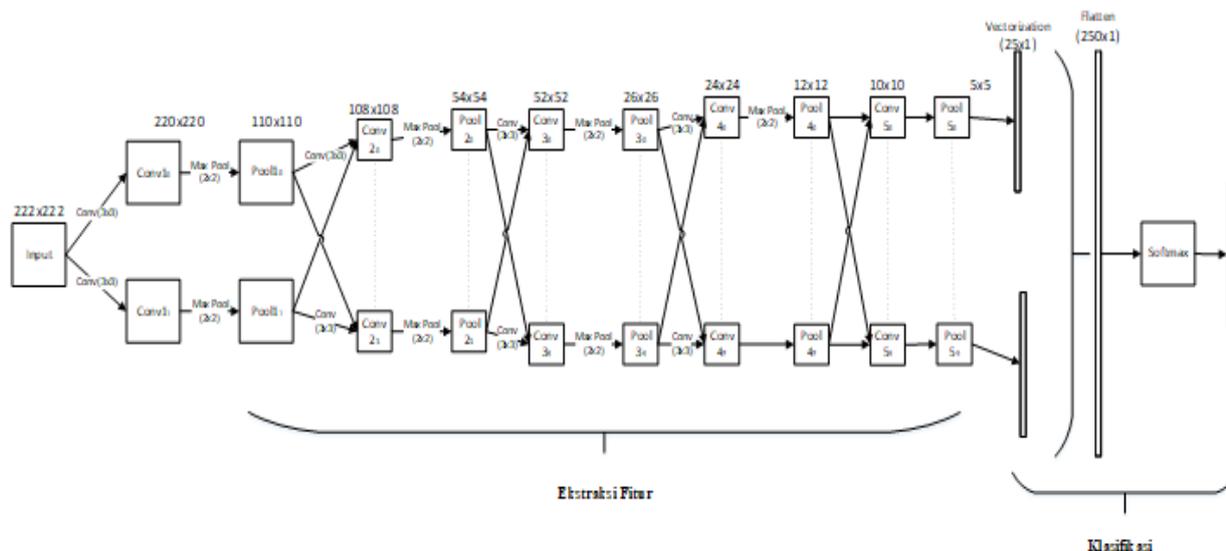
y menunjukkan posisi y baru, lb untuk menentukan ukuran lebar dari matriks baru, pp menunjukkan nilai pixel lama, dan la menunjukkan ukuran lebar matriks lama.

2.6. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu kelas *deep learning* [12]. *Deep learning* adalah bagian dari *Artificial Neural Network* [13]. Pada tahapan ini akan dilakukan proses training pada data masukan yang sebelumnya telah diolah pada tahap *preprocessing*. Tahapan-tahapan yang akan dilakukan pada proses CNN ditampilkan pada gambar 7.



Gambar 7. Tahapan CNN



Gambar 8. Arsitektur CNN

A. Inisialisasi Tahap Awal

Tahapan ini dibutuhkan untuk memenuhi tahapan selanjutnya, di mana nilai awal yang akan diinisialisasi adalah:

a. Inisialisasi parameter

Parameter *training* yang akan diinisialisasi meliputi *epoch*, *learning rate* (α), dan *error*.
 $epoch = 10$, $\alpha = 0,01$, $error = 0,00001$

b. Inisialisasi Bobot

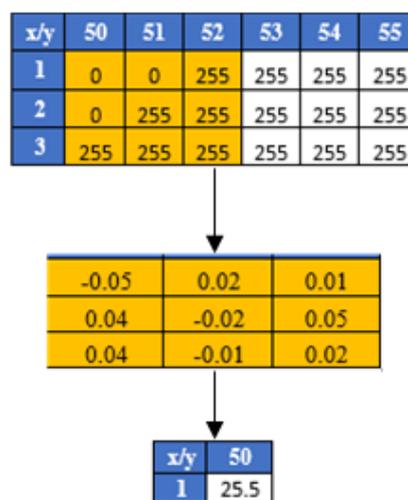
Bobot yang akan diinisialisasi adalah nilai awal bobot filter dan bias yang akan digunakan pada tahapan konvolusi dan bobot (W) yang akan digunakan pada *fully connected layer*. Untuk ukuran filter yang digunakan adalah 3×3 , dengan nilai bobot acak dari rentang $-0,05$ sampai dengan $0,05$. Kemudian nilai bobot (W) dengan nilai acak dari rentang 0 sampai dengan $0,5$.

B. Feed Forward

Pada tahapan ini nilai matriks *preprocessing* akan digunakan sebagai matriks input. Kemudian akan dilakukan proses CNN dengan arsitektur pada gambar 8. Pada penelitian ini, arsitektur CNN usulan memiliki 5 konvolusi layer, 5 pooling layer dan 1 *fully connected layer*. Pada layer konvolusi ke-1 memiliki jumlah kernel filter sebanyak 2 yang di lambangkan $C1_0$ dan $C1_1$, pada layer konvolusi ke-2 memiliki kernel filter sebanyak 4 yang dilambangkan $C2_0$ sampai dengan $C2_3$, pada layer konvolusi 3 memiliki kernel filter sebanyak 6 yang dilambangkan $C3_0$ sampai dengan $C3_5$, pada layer konvolusi ke 4 memiliki kernel filter sebanyak 8 yang dilambangkan $C4_0$ sampai dengan $C4_7$, dan pada layer layer konvolusi ke-5 memiliki kernel filter sebanyak 10 yang dilambangkan $C5_0$ sampai dengan $C5_9$, masing kernel filter berukuran 3×3 . Berikut adalah tahapan-tahapan proses berdasarkan arsitektur CNN.

a. Convolution Layer Pertama

Pada *layer* konvolusi pertama akan dilakukan operasi konvolusi antara matriks citra masukan dengan matriks kernel filter $C1_0$ filter ini akan bergeser dari pojok kiri atas citra sampai dengan pojok kanan bawah citra yang kemudian akan menghasilkan *feature map*. Pada konvolusi pertama ini filter yang digunakan berjumlah 2 filter dengan ukuran 3×3 . Adapun ilustrasi pada proses konvolusi ini telah tergambarakan pada gambar 9.



Gambar 9. Ilustrasi proses konvolusi

Adapun persamaan yang digunakan untuk tahapan ini adalah persamaan 13 dan 14.

$$Conv_{j,x,y} = (I \otimes C1_0 + B_j), \quad (13)$$

$$di\ mana \ \otimes = \sum_{u=1}^3 \sum_{v=1}^3 I_{x+u,y+v} * C1_0 \quad (14)$$

Keterangan:

$I_{x,y}$ = Nilai pixel dari input ke-x,y.

$C_{u,v}$ = Nilai pixel dari filter ke-u,v.

B_j = Nilai bias ke-j.

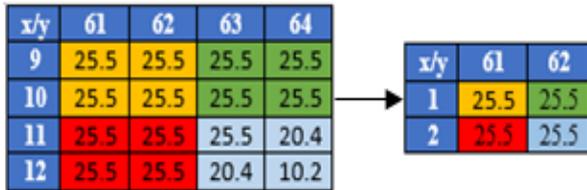
b. Aktivasi ReLU Pertama

Rectified Linear Units (ReLU) layer. Lapisan ini mengaplikasikan fungsi aktivasi $f(x) = \max(0, x)$. dimana nilai piksel yang memiliki nilai < 0 akan dirubah menjadi 0, fungsi ini meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang pada *convolutional layer*. Tujuan ReLU adalah untuk memperkenalkan nonlinearitas kepada CNN. Adapun persamaan pada tahapan ini adalah sebagai berikut.

$$f(x) = ReLU(x) = \begin{cases} C_{j,x,y} & \text{jika } (C_{j,x,y} \geq 0) \\ 0 & \text{jika tidak} \end{cases} \quad (15)$$

c. Pooling Layer Pertama

Pooling Layer berfungsi menjaga ukuran data ketika *convolution*, yaitu dengan melakukan *downsampling* atau pereduksian sampel. Dengan *pooling* kita dapat merepresentasikan data menjadi lebih kecil, mudah dikelola, dan mudah mengontrol *overfitting*. Proses *pooling* yang umum digunakan adalah *max pooling*, yaitu memilih nilai maksimum dalam suatu area tertentu, seperti diilustrasikan pada Gambar 10.



Gambar 10 Ilustrasi Max pooling

$Pool_{x,y}$

$$= \text{Max}(Conv_{x,y}, Conv_{x+1,y}, Conv_{x,y+1}, Conv_{x+1,y+1}) \quad (16)$$

d. Convolution Layer Kedua

Pada layer konvolusi kedua ini perhitungan yang dilakukan sedikit berbeda dimana inputan dari layer sebelumnya sebanyak jumlah filter masing-masing akan dikonvolusi dan kemudian akan dijumlahkan. Persamaan 17 digunakan pada tahapan layer konvolusi kedua.

$$Conv_j = \sum_{i=1}^{filter} pool_i \otimes C_{i,j} + B_j \quad (17)$$

Keterangan:

$Conv_j$ = Layer konvolusi ke-j.

$pool_i$ = Layer *pooling* ke-i.

$C_{i,j}$ = Kernel index ke-j dan filter ke-i.

B_j = Bias ke-j.

\otimes = Operasi konvolusi.

e. Fully Connected Layer

Pada lapisan yang terhubung secara penuh (*fully connected layer*), setiap *neurons* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya. Hal ini sama persis dengan yang ada pada MLP. Model aktivasinya pun sama persis dengan MLP [12]. *Fully connected layer* berperan untuk mengklasifikasi data masukan. Output yang dihasilkan dari *pooling layer* masih berbentuk *multidimensional array*, sehingga akan di *flatten* terlebih dahulu untuk mengubah data menjadi vektor sebelum input untuk *fully connected layer*. Adapun persamaan yang digunakan pada tahapan ini adalah sebagai berikut.

$$Conv_j = \sum_{i=1}^{filter} pool_i \otimes C_{i,j} + B_j \quad (18)$$

Keterangan:

$Fully_i$ = Hasil dari perhitungan pada *fully-connected layer*.

$W_{i,j}$ = Nilai bobot yang digunakan dari hasil *convolutional layer*.

$flatten_j$ = Nilai dari vektor ke-j.

i = Kelas ke-i ($i = 1,2,3,4,5,6,7$).

f. Softmax

Kemudian hasil dari *Fully* akan diaktifasi dengan menggunakan fungsi *softmax*, digunakan fungsi *softmax* karena pada penelitian ini menghasilkan *output* atau jumlah kelas lebih dari dua [14], atau biasa disebut dengan *multi-class*. Tujuan dari fungsi *softmax* sendiri adalah agar dapat diketahui prediksi yang dihasilkan dari arsitektur yang dibangun, dimana rumus untuk fungsi *softmax* ditulis pada persamaan 19.

$$\hat{y}_i = \frac{e^{Fully_i}}{(\sum_{j=1}^{kelas} e^{Fully_j})} \quad (19)$$

Keterangan:

\hat{y}_i = Hasil dari aktifasi fungsi *softmax*.

$Fully_i$ = Hasil dari perhitungan pada *fully-connected layer* ke-i.

$flatten_j$ = Nilai dari vektor ke-j.

g. Cross-Entropy Loss Function

Selanjutnya akan dicari nilai *error* yang didapat dari hasil prediksi pada *fully-connected layer* sebelumnya, dimana nilai *error* ini digunakan untuk menentukan apakah hasil prediksi dari CNN sudah mencapai target atau belum, maka dari itu akan dilakukan perbandingan antara *error* dengan *target error* dimana, apabila *error* masih dibawah dari target, maka akan dilakukan *loss function*. Dari beberapa jenis *loss function* yang ada, salah satunya adalah

Cross-entropy Loss Function, dimana rumus dari cross-entropy tersebut dituliskan pada persamaan 20.

$$Loss = - \sum_i^{kelas} t_i * Log(\hat{y}) \quad (20)$$

C. Backpropagation

Backpropagation adalah salah satu algoritma supervised learning yang digunakan dalam artificial neural networks. Backpropagation mencari kombinasi bobot untuk meminimalkan kesalahan output untuk dianggap menjadi solusi yang benar [15]. Adapun tahapan Backpropagation adalah sebagai berikut:

a. Turunan gradien error terhadap softmax

Tahapan turunan gradien error terhadap softmax berdasarkan rumus yang ditulis oleh Peter Sadowski [15].

$$\Delta \hat{y}_i = \frac{\partial Loss}{\partial \hat{y}_i} = \hat{y}_i - t_i \quad (21)$$

b. Turunan gradien error terhadap bobot

Tahapan turunan gradien error terhadap bobot berdasarkan rumus yang ditulis oleh Zhifei Zhang [14].

$$\Delta W_{i,j} = \frac{\partial Loss}{\partial W_{i,j}} = \Delta \hat{y}_i * flatten_j \quad (21)$$

c. Turunan gradien error terhadap bias

Tahapan turunan gradien error terhadap bobot

$$\Delta b_i = \frac{\partial Loss}{\partial b_i} = \Delta \hat{y}_i \quad (22)$$

D. Stochastic Gradient Descent

Stochastic Gradient Descent adalah sebuah algoritma untuk menemukan nilai minimum lokal dari sebuah fungsi, Algoritma dari Stochastic Gradient Descent dirumuskan untuk nilai perbaikan bobot dan bias baru seperti pada persamaan berikut:

a. Perbaikan nilai pada bobot

$$\theta W = W - \alpha(\Delta W) \quad (23)$$

Keterangan:

ΔW = Nilai turunan dari bobot.

W = Nilai bobot.

α = Nilai learning rate.

θW = Nilai bobot baru.

b. Perbaikan nilai pada bias

$$\theta b = b - \alpha(\Delta b) \quad (24)$$

Keterangan:

Δb = Nilai turunan dari bias.

b = Nilai bias.

α = Nilai learning rate.

θb : Nilai bias baru.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Data

Pada penelitian ini data yang akan digunakan adalah data gambar yang dibagi menjadi dua jenis yaitu data latih dan data uji. Sumberdata dilakukan dengan menangkap ekspresi wajah secara langsung terhadap 42 orang responden. Data latih dan data uji diambil dari 42 orang sebagai subjek 21 orang berekspresi bahagia, sedih terkejut, takut, marah, muak dan 21 orang lainnya berekspresi netral. Masing-masing data diambil sebanyak 15 citra per orang. Total jumlah data keseluruhan adalah 2205.

3.2. Pengujian Akurasi

Pengujian akurasi pada penelitian ini menggunakan metode confusion matrix. Pada dasarnya confusion matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan klasifikasi yang seharusnya. Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 istilah sebagai representasi hasil proses klasifikasi. Keempat istilah ini adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Parameter pengujian yang digunakan pada penelitian ini berdasarkan penelitian yang dilakukan oleh Shiyang Zhou [6].

A. Pengujian dengan segmentasi

Adapun hasil terbaik yang didapat pada epoch 1000 dan learning rate 0,0005 ditunjukkan pada tabel 1.

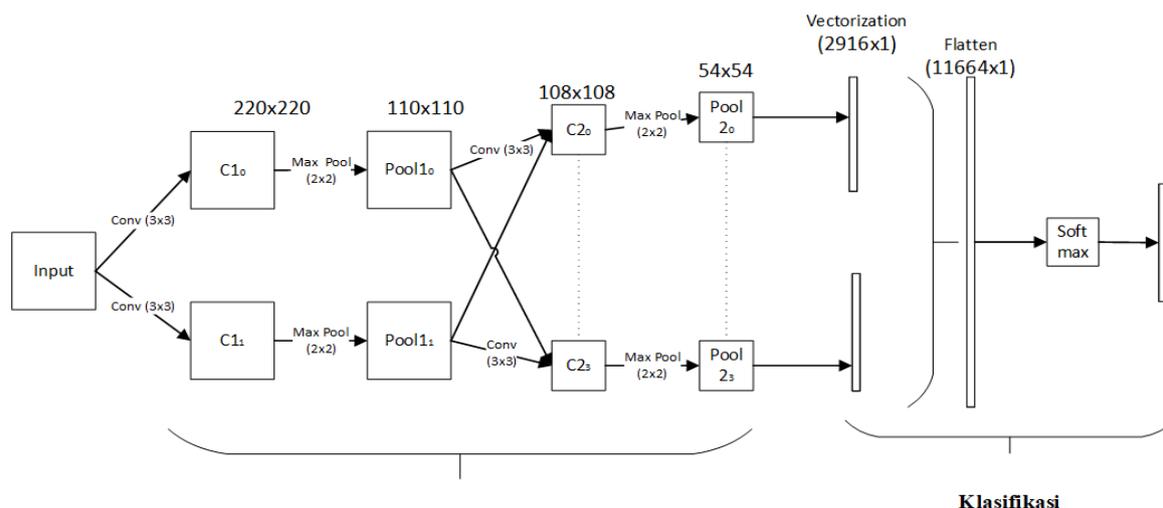
Tabel 1. Pengujian dengan segmentasi

| Kelas sebenarnya | Hasil Prediksi | | | | | | |
|------------------|----------------|-----|-----|-----|-----|-----|-----|
| | (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| Bahagia (1) | 47 | 6 | 0 | 6 | 0 | 4 | 0 |
| Marah (2) | 6 | 29 | 4 | 3 | 2 | 0 | 10 |
| Muak (3) | 9 | 5 | 37 | 0 | 6 | 3 | 2 |
| Netral (4) | 0 | 0 | 0 | 63 | 0 | 0 | 0 |
| Sedih (5) | 9 | 6 | 3 | 0 | 19 | 11 | 12 |
| Takut (6) | 12 | 5 | 7 | 3 | 1 | 26 | 9 |
| Terkejut (7) | 6 | 6 | 2 | 0 | 2 | 10 | 37 |

Akurasi rata-rata = 58,50%

B. Pengujian tanpa segmentasi

Adapun hasil terbaik yang didapat pada epoch 1000 dan learning rate 0,001 ditunjukkan pada tabel 2.



Ekstraksi Fitur

Gambar 11. Arsitektur CNN 2 Layer

Tabel 2. Pengujian tanpa segmentasi

| Kelas sebenarnya | Hasil Prediksi | | | | | | |
|------------------|----------------|-----|-----|-----|-----|-----|-----|
| | (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| Bahagia (1) | 28 | 0 | 4 | 0 | 0 | 31 | 0 |
| Marah (2) | 6 | 37 | 8 | 0 | 0 | 12 | 0 |
| Muak (3) | 8 | 0 | 4 | 3 | 28 | 17 | 3 |
| Netral (4) | 4 | 5 | 27 | 10 | 0 | 16 | 1 |
| Sedih (5) | 13 | 8 | 1 | 5 | 21 | 8 | 7 |
| Takut (6) | 0 | 0 | 21 | 0 | 1 | 40 | 1 |
| Terkejut (7) | 4 | 8 | 5 | 0 | 3 | 13 | 30 |

Akurasi rata-rata = 38,54%

Akurasi yang didapat masih terlalu kecil, setelah dilakukan evaluasi kembali terhadap arsitektur yang dibangun, terdapat masalah yaitu banyak *feature map* hasil konvolusi yang bernilai 0, hal ini mengakibatkan informasi yang seharusnya digunakan untuk klasifikasi banyak yang hilang. Oleh karena itu akan dilakukan pengujian kembali menggunakan arsitektur yang lain. Pengujian arsitektur yang berbeda ini dilakukan karena menurut Josh Tobin kecilnya akurasi pada *deep learning* salah satunya adalah karena arsitektur *deep learning* yang kurang optimal. Selanjutnya digunakan layer yang tidak terlalu banyak seperti penggunaan 2 *convolutional layer*, 2 *max pooling layer* dan 1 *fully connected layer*. Oleh karena itu akan dilakukan pengujian kembali dengan menggunakan 2 *convolutional layer*, 2 *max pooling layer* dan 1 *fully connected layer*. Gambar 11 adalah arsitektur yang dibangun kembali.

C. Pengujian arsitektur 2 layer dengan segmentasi

Adapun hasil terbaik yang didapat pada *epoch* 10 dan *learning rate* 0,001 ditunjukkan pada tabel 3.

| Kelas sebenarnya | Hasil Prediksi | | | | | | |
|------------------|----------------|-----|-----|-----|-----|-----|-----|
| | (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| Bahagia (1) | 60 | 0 | 0 | 3 | 0 | 0 | 0 |
| Marah (2) | 0 | 63 | 0 | 0 | 0 | 0 | 0 |
| Muak (3) | 0 | 0 | 63 | 0 | 0 | 0 | 0 |
| Netral (4) | 0 | 0 | 0 | 63 | 0 | 0 | 0 |
| Sedih (5) | 0 | 1 | 1 | 0 | 53 | 2 | 2 |
| Takut (6) | 1 | 0 | 0 | 0 | 3 | 59 | 0 |
| Terkejut (7) | 1 | 0 | 0 | 0 | 0 | 0 | 63 |

Akurasi rata-rata = 96,14%

D. Pengujian arsitektur 2 layer tanpa segmentasi

Adapun hasil terbaik yang didapat pada *epoch* 25 dan *learning rate* 0,0005 ditunjukkan pada tabel 4.

Tabel 3. Pengujian dengan segmentasi

| Kelas sebenarnya | Hasil Prediksi | | | | | | |
|------------------|----------------|-----|-----|-----|-----|-----|-----|
| | (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| Bahagia (1) | 38 | 6 | 8 | 0 | 6 | 0 | 0 |
| Marah (2) | 4 | 48 | 8 | 0 | 3 | 0 | 0 |
| Muak (3) | 0 | 9 | 48 | 3 | 3 | 0 | 0 |
| Netral (4) | 0 | 3 | 0 | 60 | 0 | 0 | 0 |
| Sedih (5) | 4 | 1 | 1 | 2 | 52 | 1 | 2 |
| Takut (6) | 0 | 0 | 3 | 13 | 11 | 36 | 8 |
| Terkejut (7) | 0 | 0 | 0 | 4 | 0 | 3 | 56 |

Akurasi rata-rata = 76,64%

3.3. Pembahasan Hasil Pengujian

Setelah dilakukan pengujian dengan arsitektur yang berbeda berhasil mendapatkan akurasi terbaik sebesar 96,14% dengan menggunakan segmentasi, sedangkan tanpa segmentasi berhasil mendapatkan akurasi terbaik sebesar 76,64% . Dari hasil pengujian dengan menggunakan proses segmentasi pada proses klasifikasi ekspresi wajah adanya peningkatan akurasi sebesar 19,96% pada pengujian dengan arsitektur yang diajukan, kemudian adanya peningkatan akurasi juga pada pengujian dengan arsitektur yang berbeda yaitu sebesar 18,14%

4. KESIMPULAN

Berdasarkan hasil pengujian dengan *confusion matrix* yang telah dilakukan, sesuai dengan arsitektur CNN yang diusulkan dengan parameter *epoch* 10 dan *learning rate* 0,001 diperoleh akurasi sebesar 96,14%. Terdapat beberapa temuan yang diperoleh yaitu :

1. Akurasi yang kecil disebabkan karena banyaknya informasi yang hilang pada *feature map* hasil dari proses konvolusi ke-5.
2. Penggunaan segmentasi *Viola Jones* pada saat proses segmentasi dapat menghasilkan peningkatan akurasi yang cukup signifikan

5. DAFTAR PUSTAKA

- [1] M. S. Zia and M. A. Jaffar, "An adaptive training based on classification system for patterns in facial expressions using SURF descriptor templates," *Multimed. Tools Appl.*, vol. 74, no. 11, pp. 3881-3899, 2015.
- [2] A. Mehrabian, "Silent Messages: Implicit Communication of Emotions and Attitudes," *Child. Educ.*, p. 196, 1981.
- [3] P. Rathod, L. Gagnani, and K. Patel, "Facial Expression Recognition: Issues and Challenges," *Int. J. Enhanc. Res. Sci. Technol. Eng.*, vol. 3, no. 2, pp. 108-111, 2014.
- [4] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2015-October, pp. 34-42, 2015.
- [5] D. Jaswal, S. V. Soman, and K. P. Soman, "Image Classification Using Convolutional Neural Networks," *Int. J. Sci. Eng. Res.*, vol. 5, no. 6, pp. 1661-1668, 2014.
- [6] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, "Classification of surface defects on steel sheet using convolutional neural networks," *Mater. Technol.*, vol. 51, no. 1, pp. 123-131, 2017.
- [7] G. Antipov, S. A. Berrani, and J. L. Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," *Pattern Recognit. Lett.*, vol. 70, pp. 59-65, 2016.
- [8] L. Zheng and S. Zhu, "Convolutional Neural Network for Facial Expression Recognition," *Conf. Neural Inf. Process. Syst. (NIPS 2016)*, vol. 2018-Januari, 2016.
- [9] Priyanto, *Pengolahan Citra Digital Teori dan Aplikasi*. Bandung: Informatika, 2017.
- [10] V. Hlavac, "Fundamentals of Image Processing," *Opt. Digit. Image Process. Fundam. Appl.*, no. November 2012, pp. 71-96, 2011.
- [11] M. Viola, Paul; Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*. CVPR 2001. Vol. 1. IEEE, 2001.
- [12] Suyanto, *Machine Learning Tingkat Dasar Dan Lanjut*. Bandung: Informatika, 2018.
- [13] N. I. Widiastuti, "Deep Learning - Now and Next in Text Mining and Natural Language Processing," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018.
- [14] Z. Zhang, "Derivation of Backpropagation in Convolutional Neural Network (CNN)," *University of Tennessee, Knoxville, TN*, 2016.
- [15] M. Sapkota, X. Shi, F. Xing, and L. Yang, "Deep Convolutional Hashing for Low-Dimensional Binary Embedding of Histopathological Images," *IEEE J. Biomed. Heal. Informatics*, vol. 23, no. 2, pp. 805-816, Mar. 2019.