

Pengenalan Emosi Menggunakan ekspresi Wajah dengan Metode CNN

Harith Fakhiri Setiawan - 13519161
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): harithfakhiri@gmail.com

Abstraksi—*Convolutional Neural Network (CNN)* merupakan salah satu arsitektur *deep learning* yang mempelajari tentang pembelajaran terhadap suatu citra. Dengan dikembangkannya CNN, pengenalan, klasifikasi, ataupun pembelajaran lainnya terhadap berbagai objek visual dapat dilakukan. Salah satu contohnya pengenalan emosi yang tadinya hanya dapat dilakukan oleh manusia, sekarang ini dapat dilakukan juga oleh CNN.

Keywords—*CNN; Deep learning; Pembelajaran mesin; Emosi; Model; Citra;*

I. PENDAHULUAN

Emosi manusia adalah hal yang dapat dikenali dengan berbagai cara, mulai dari ekspresi atau mimik wajah, intonasi berbicara, ataupun tingkah laku manusia lainnya yang dapat dirasakan melalui indra dan diinterpretasikan dengan bantuan otak manusia. Kemampuan interpretasi ini membuat manusia dapat memahami satu sama lain dengan lebih baik dengan manusia lainnya.

Perkembangan teknologi telah menuntun manusia untuk melangkah lebih jauh dalam mengeksplorasi berbagai hal, salah satunya otomasi. Otomasi membuat pekerjaan yang awalnya hanya dapat dikerjakan manusia dapat juga dikerjakan oleh mesin. Seiring berjalannya waktu, otomasi yang tadinya hanya mampu mengerjakan hal yang berbasis instruksi sekarang ini juga dapat mengerjakan hal yang berbasis intelijen. Teknologi ini sering juga dikenal dengan *Artificial Intelligence (AI)* atau kecerdasan buatan. Salah satu bentuk dari kecerdasan buatan ini adalah *machine learning* atau pembelajaran mesin, yaitu kemampuan mesin untuk mempelajari berbagai hal dengan meniru cara kerja otak manusia. Dalam pengembangannya, pembelajaran mesin memiliki cabang yang dikenal dengan *deep learning*, yang terfokus dalam melakukan pembelajaran berdasarkan pengalaman.

Dengan adanya pembelajaran mesin, pengenalan emosi manusia dengan bantuan mesin tidak lagi menjadi hal yang mustahil. Salah satu metode yang dapat dilakukan oleh pembelajaran mesin adalah pengenalan emosi melalui ekspresi wajah dengan bantuan *Convolutional Neural Network (CNN)* yang meniru koneksi neuron pada sel saraf otak manusia. CNN terinspirasi dari *Visual Cortex* pada otak manusia yang memiliki tugas untuk memproses informasi yang diterima otak

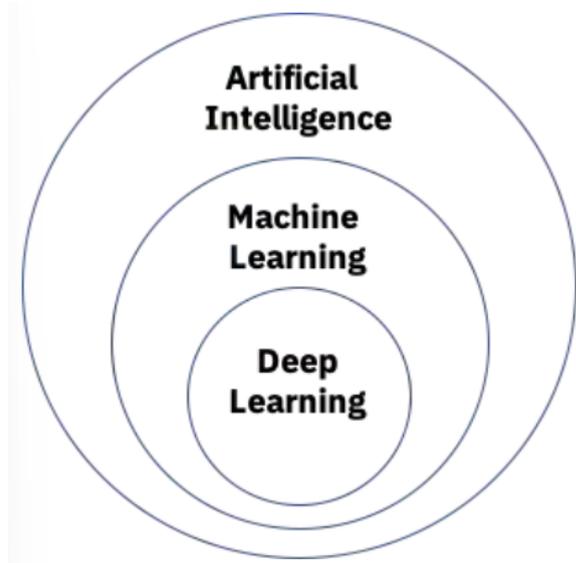
dalam bentuk visual. Untuk dapat melakukan proses pengenalan emosi, diperlukan citra digital yang di dalamnya terdapat objek wajah yang dapat dikenali.

II. LANDASAN TEORI

A. Artificial Intelligence

Menurut John McCarthy pada tahun 2004, *Artificial Intelligence* atau kecerdasan buatan adalah sebuah ilmu pengetahuan dan teknik dari pembuatan mesin yang pintar, terutama kepintaran program komputer. Kecerdasan buatan terkait dengan tugas serupa yang menggunakan komputer untuk memahami kecerdasan manusia, tanpa harus membatasi diri pada metode yang dapat diamati secara biologis^[6].

Keberadaan kecerdasan buatan membuat sistem untuk dapat berpikir dan bertindak baik seperti manusia, maupun secara ideal dalam menyelesaikan sebuah permasalahan. Kecerdasan buatan mengombinasikan ilmu komputer dengan kumpulan data untuk dapat menyelesaikan sebuah permasalahan.



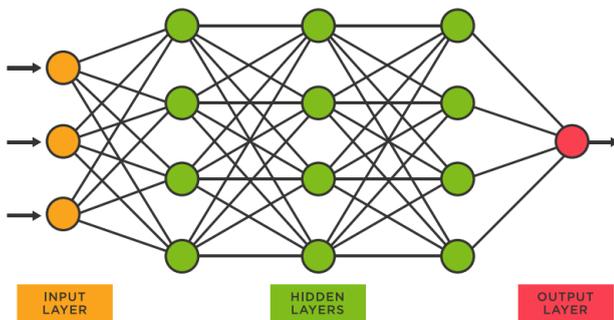
Gambar 2.1. Ranah kecerdasan buatan^[6]

B. Machine Learning

Machine learning atau pembelajaran mesin merupakan cabang dari kecerdasan buatan dan ilmu komputer yang memiliki fokus pada pemanfaatan sebuah kumpulan data dan algoritma untuk meniru cara manusia belajar untuk menyelesaikan sebuah permasalahan^[7]. Dengan pemanfaatan sebuah dataset dan algoritma tertentu, sebuah pembelajaran dapat terus melakukan perbaikan secara terus menerus untuk meningkatkan akurasi dari prediksi yang dibuat oleh mesin tersebut.

C. Deep Learning

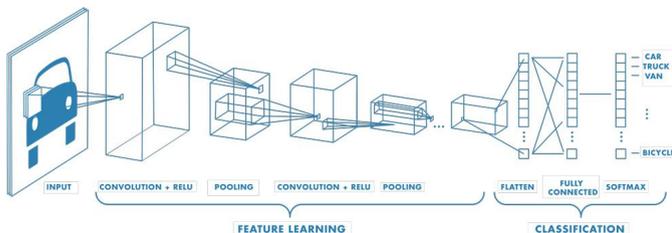
Deep learning merupakan salah satu dari jenis dari pembelajaran mesin yang pembelajaran layaknya manusia pada umumnya, yaitu belajar dari *experience* atau pengalaman. Pada proses pembelajarannya, *deep learning* menggunakan konsep peniruan jaringan syaraf atau *artificial neural networks* untuk mempelajari dan mengenali berbagai representasi yang berguna dari fitur yang diambil dari informasi atau data yang diterimanya. Umumnya, *deep learning* melakukan pembelajaran terhadap informasi dalam bentuk suara, gambar, ataupun teks.



Gambar 2.2. Artificial Neural Network^[3]

D. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu algoritma *deep learning* yang umumnya digunakan untuk melakukan pembelajaran terhadap citra digital. Pembelajaran ini dilakukan dengan melakukan ekstraksi fitur-fitur penting secara otomatis yang terdapat pada sebuah citra visual untuk mempelajari suatu pola tertentu dengan tujuan menyimpulkan informasi yang dibutuhkan. CNN umumnya digunakan untuk melakukan pengenalan objek, klasifikasi objek, dan bentuk pembelajaran lainnya yang dapat diambil dari sebuah citra digital. Berikut adalah gambaran umum dari arsitektur CNN.

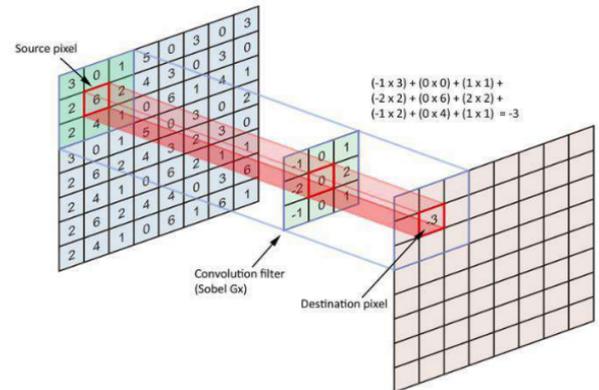


Gambar 2.3. Arsitektur CNN^[4]

Terdapat 3 layer atau lapisan utama pada CNN, yaitu:

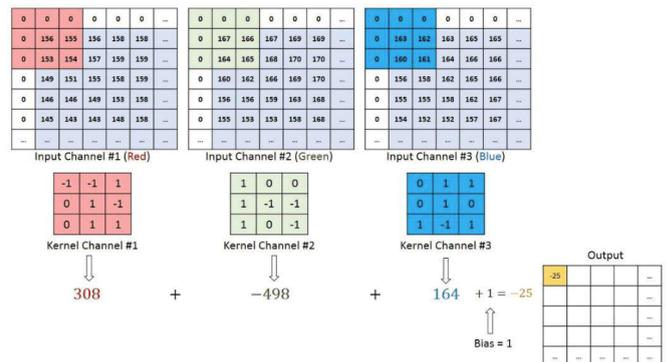
1. Convolution Layer + (Batch Normalization & ReLU)

Convolutional Layer atau lapisan konvolusi adalah lapisan dimana citra masukan dioperasikan secara konvolusi dengan matriks *filter*. Ukuran dari matriks *filter* disesuaikan dengan kebutuhan Luaran dari proses ini disebut dengan *feature map*.



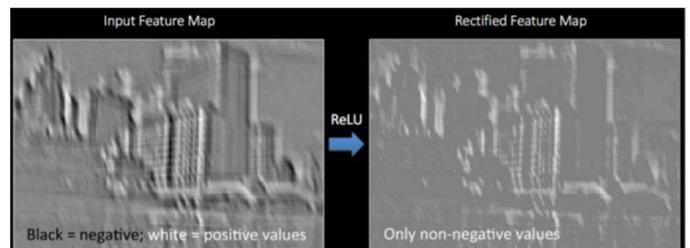
Gambar 2.4. Operasi konvolusi pada *convolutional layer*^[3]

Operasi konvolusi dengan matriks *filter* yang dilakukan pada citra input dilakukan pada setiap kanal yang terdapat dalam citra. Sebagai contoh, apabila citra masukan merupakan citra RGB maka konvolusi dengan matriks *filter* dilakukan untuk kanal merah, hijau, dan biru.



Gambar 2.5. operasi konvolusi pada citra RGB^[3]

Setelah konvolusi dilakukan, terdapat lapisan tambahan dengan melakukan aktivasi fungsi yang dikenal dengan *Rectified Linear Unit (ReLU)*. Pada lapisan ini, elemen matriks pada *feature map* yang bernilai negatif dipetakan ke nol.



Gambar 2.6. ReLU^[3]

Di antara *convolutional layer* dan ReLU, terdapat lapisan yang dikenal dengan *Batch Normalization Layer* yang berfungsi untuk menormalisasi aktivasi dan juga gradient yang terdapat pada jaringan. Normalisasi ini bertujuan untuk membuat pembelajaran menjadi permasalahan optimasi yang lebih mudah.

2. Pooling Layer

Pooling layer adalah lapisan yang bertanggung jawab untuk mengurangi ukuran spasial dari *feature maps* setelah konvolusi yang bertujuan mengurangi dimensi untuk memperkecil daya komputasi yang diperlukan dalam pemrosesan data.

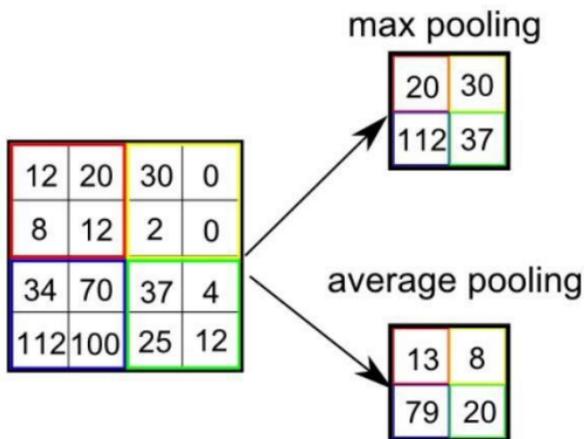
Terdapat dua jenis *pooling*, yaitu:

a. Max pooling

Metode ini mengembalikan nilai maksimum dari bagian piksel sesuai dengan ukuran kernel.

b. Average pooling

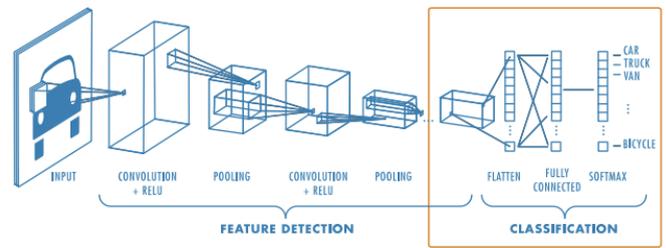
Metode ini mengembalikan nilai rerata semua elemen dari bagian piksel sesuai dengan ukuran kernel.



Gambar 2.7. Pooling^[8]

3. Fully-connected layer

Pada lapisan ini, dihasilkan vektor berdimensi K, dimana K merepresentasikan jumlah kelas yang dapat diprediksi oleh jaringan. Pada vektor ini, dilakukan generasi probabilitas kemungkinan pengklasifikasian dari setiap kelas yang ada dengan bantuan fungsi *softmax*. Luaran dari vektor yang dihasilkan dari fungsi *softmax* ini digunakan untuk melakukan klasifikasi dengan memilih kelas yang sesuai dengan citra masukan.



Gambar 2.8. klasifikasi pada *fully-connected layer*^[4]

E. Citra Digital

Citra atau yang biasa dikenal dengan nama gambar merupakan sinyal dwimitra (2-D) yang bersifat kontinu secara visual dan dapat diamati oleh sistem visual yang dimiliki manusia^[1]. Dengan demikian, citra digital adalah citra yang sudah terkonversi dalam bentuk digital. Citra digital yang tersimpan dalam bentuk digital juga dapat divisualisasikan.



Gambar 2.9. Contoh Citra Digital^[1]

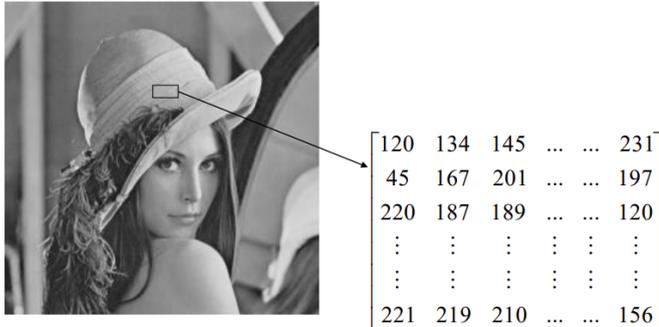
Citra digital direpresentasikan menggunakan matriks berukuran M x N, dimana setiap elemen matriksnya menyatakan nilai dari setiap piksel dan M x N menyatakan resolusi citra. Elemen atau piksel ini direpresentasikan dalam dalam koordinat x,y secara diskrit, dimana elemen pikselnya dinyatakan dalam fungsi $f(x,y)$ yang menyatakan intensitas cahaya yang terdapat di koordinat (x,y). Oleh karena itu, walaupun secara visual citra bersifat kontinu, dalam representasi sebenarnya elemen citra disimpan dalam kumpulan piksel yang bersifat diskrit.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

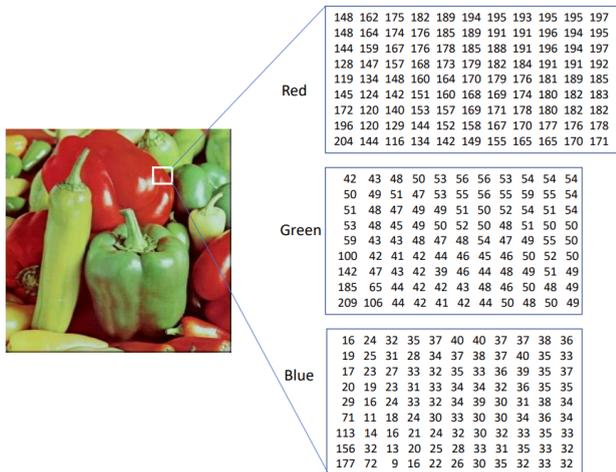
Gambar 2.10. Representasi citra digital^[1]

Dalam matriks yang merepresentasikan citra digital, setiap piksel merepresentasikan *channel* atau kanal dalam bentuk yang berbeda-beda. Perbedaan bentuk piksel ini bergantung kepada jenis-jenis citra yang disimpan yang mempengaruhi

kedalaman setiap pikselnya. Sebagai contoh, citra *grayscale* hanya memiliki kedalaman bernilai 1, sehingga direpresentasikan dalam bentuk $M \times N$. Sementara citra berwarna dapat direpresentasikan dalam bentuk $M \times N \times b$, dimana b merupakan kedalaman. Sebagai contoh, citra RGB, yang disimpan dalam kanal merah, hijau, dan biru, memiliki kedalaman bernilai 3. Berikut adalah perbandingan representasi citra *grayscale* dan citra RGB.



Gambar 2.11. Representasi citra *grayscale* [1]



Gambar 2.12. Representasi citra RGB [2]

III. IMPLEMENTASI

Solusi dari permasalahan pengenalan emosi berdasarkan ekspresi wajah dapat diselesaikan menggunakan pengimplementasian metode *Convolutional Neural Network* (CNN).

A. Pengumpulan dataset

Pengimplementasian *deep learning* menggunakan CNN dapat dilakukan dengan melakukan pembelajaran terhadap sekumpulan data yang relevan dengan permasalahan. Oleh karena itu, digunakan *public dataset* yang CKPLUS. *Dataset* ini berisi berbagai ekspresi wajah seseorang dapat diklasifikasikan menjadi 7 emosi, yaitu *anger*, *contempt*, *disgust*, *fear*, *happy*, *sadness*, dan *surprise*.



Gambar 3.1. emosi *anger*, *contempt*, *disgust*, *fear*, *happy*, *sadness*, dan *surprise*.

B. Pra-pemrosesan citra

Salah satu tujuan melakukan pra-pemrosesan adalah untuk mengoptimalkan proses ataupun hasil prediksi yang ingin diselesaikan oleh program. Oleh karena itu, dilakukan *resizing* atau pengubahan nilai ukuran citra menjadi 48 x 48. Pengubahan ukuran ini bertujuan untuk mengoptimalkan daya komputasi agar program dapat bekerja dengan lebih cepat. Pengubahan ukuran ini diimplementasikan menggunakan fungsi *resize* pada library *opencv*.

```
input_img_resize=cv2.resize(input_img, (48, 48))
```

C. Pendekatan Menggunakan Convolutional Neural Network

Dalam pengimplementasian algoritma CNN, dilakukan beberapa langkah pengerjaan sebagai berikut.

1. Pembagian dataset

dilakukan pembagian terhadap *dataset* menjadi data latih dan data tes. Pembagian dibagi dengan rasio 8:2. Sebelum dilakukan pembagian, data di *shuffle* terlebih dahulu untuk memastikan persebaran data merata.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
```

2. Pembuatan model CNN

Model CNN yang digunakan dibuat menggunakan *open-source library* yang dikenal dengan *tensorflow*. *Layer* atau lapisan dari model CNN yang dibuat sebagai berikut.

```
model = tf.keras.Sequential()
model.add(Conv2D(32, (3, 3), padding='same', kernel_initializer='he_normal', input_shape=(48, 48, 3)))
model.add(Activation('relu'))
```

```

model.add(BatchNormalization())
model.add(Conv2D(32, (3, 3), padding='same',
kernel_initializer='he_normal', input_shape=(48, 48, 3)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

```

```

model.add(Conv2D(64, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

```

```

model.add(Conv2D(128, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

```

```

model.add(Conv2D(256, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(256, (3, 3), padding='same',
kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

```

```

model.add(Flatten())
model.add(Dense(64, kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

```

```

model.add(Dense(64, kernel_initializer='he_normal'))
model.add(Activation('relu'))

```

```

model.add(BatchNormalization())
model.add(Dropout(0.5))

```

```

model.add(Dense(7, kernel_initializer='he_normal'))
model.add(Activation('softmax'))

```

```

Model: "sequential_1"

```

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 48, 48, 32)	896
activation_11 (Activation)	(None, 48, 48, 32)	0
batch_normalization_10 (Batch Normalization)	(None, 48, 48, 32)	128
conv2d_9 (Conv2D)	(None, 48, 48, 32)	9248
activation_12 (Activation)	(None, 48, 48, 32)	0
batch_normalization_11 (Batch Normalization)	(None, 48, 48, 32)	128
max_pooling2d_4 (Max Pooling 2D)	(None, 24, 24, 32)	0
dropout_6 (Dropout)	(None, 24, 24, 32)	0
conv2d_10 (Conv2D)	(None, 24, 24, 64)	18496
activation_13 (Activation)	(None, 24, 24, 64)	0
batch_normalization_12 (Batch Normalization)	(None, 24, 24, 64)	256
conv2d_11 (Conv2D)	(None, 24, 24, 64)	36928
activation_14 (Activation)	(None, 24, 24, 64)	0
batch_normalization_13 (Batch Normalization)	(None, 24, 24, 64)	256
max_pooling2d_5 (Max Pooling 2D)	(None, 12, 12, 64)	0
dropout_7 (Dropout)	(None, 12, 12, 64)	0
conv2d_12 (Conv2D)	(None, 12, 12, 128)	73856
activation_15 (Activation)	(None, 12, 12, 128)	0

batch_normalization_14 (Batch Normalization)	(None, 12, 12, 128)	512
conv2d_13 (Conv2D)	(None, 12, 12, 128)	147584
activation_16 (Activation)	(None, 12, 12, 128)	0
batch_normalization_15 (Batch Normalization)	(None, 12, 12, 128)	512
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_8 (Dropout)	(None, 6, 6, 128)	0
conv2d_14 (Conv2D)	(None, 6, 6, 256)	295168
activation_17 (Activation)	(None, 6, 6, 256)	0
batch_normalization_16 (Batch Normalization)	(None, 6, 6, 256)	1024
conv2d_15 (Conv2D)	(None, 6, 6, 256)	590080
activation_18 (Activation)	(None, 6, 6, 256)	0
batch_normalization_17 (Batch Normalization)	(None, 6, 6, 256)	1024
max_pooling2d_7 (MaxPooling2D)	(None, 3, 3, 256)	0
dropout_9 (Dropout)	(None, 3, 3, 256)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_3 (Dense)	(None, 64)	147520
activation_19 (Activation)	(None, 64)	0
batch_normalization_18 (Batch Normalization)	(None, 64)	256
dropout_10 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 64)	4160
activation_20 (Activation)	(None, 64)	0
batch_normalization_19 (Batch Normalization)	(None, 64)	256
dropout_11 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 7)	455
activation_21 (Activation)	(None, 7)	0
=====		
Total params: 1,328,743		
Trainable params: 1,326,567		
Non-trainable params: 2,176		

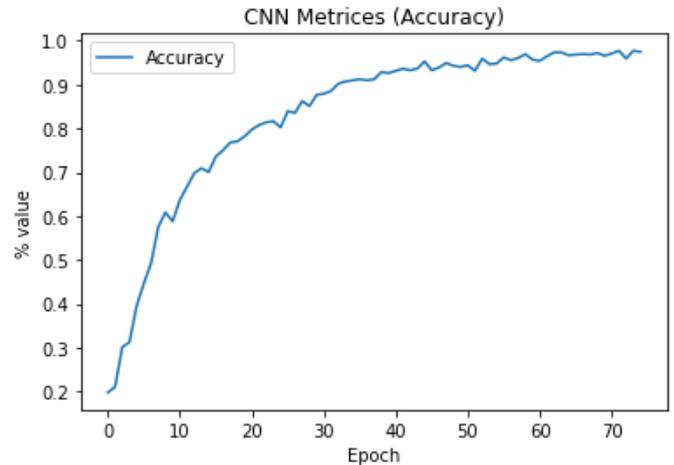
Gambar 3.2. Model Summary

Setelah itu, model dilatih dengan menggunakan *optimizer Stochastic Gradient Descent (SGD)*, yaitu adam, dengan matriks penilaian berupa akurasi dan *learning rate* sebesar 0.0005.

```
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.0005), loss='categorical_crossentropy', metrics=['accuracy'])
history = model.fit(
    x=X_train,
```

```
y=y_train,
epochs=75)
```

Berdasarkan model yang telah dibuat, mesin berhasil memiliki akurasi sebesar 97.45%.



Gambar 3.3. Grafik matriks akurasi pada setiap epoch

D. Evaluasi model

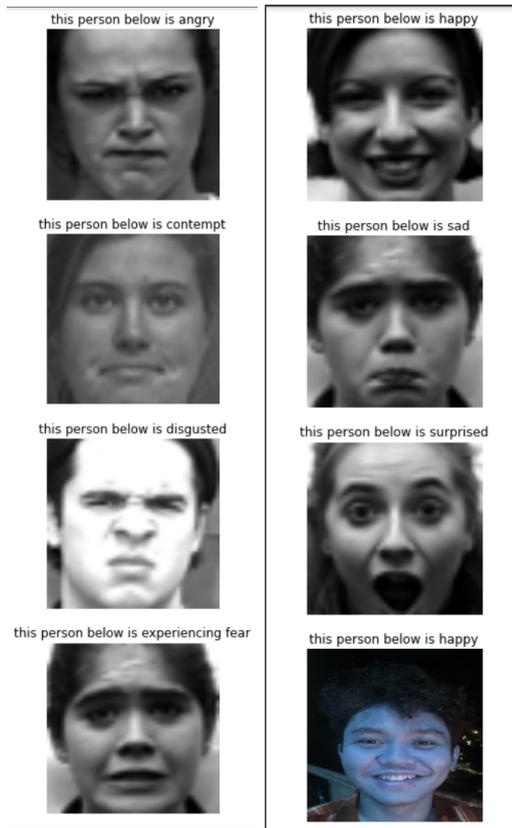
Evaluasi model dilakukan dengan mengukur akurasi dengan data tes yang sebelumnya telah dibagi. Berdasarkan pengujian data tes tersebut pada model yang telah dibuat, didapat akurasi 94.42%.

```
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test_labels, y_pred_labels)

[43] print(a)

0.9441624365482234
```

Gambar 3.4. Akurasi data tes



IV. KESIMPULAN

Convolutional Neural Network (CNN) yang merupakan salah satu arsitektur pada *deep learning* yang dapat membantu dalam pemrosesan citra visual.

Berdasarkan percobaan yang dilakukan, terbukti bahwa CNN dapat digunakan untuk melakukan pengenalan emosi berdasarkan ekspresi wajah. Walaupun demikian, masih terdapat kekurangan pada beberapa prediksi emosi yang dapat dikatakan ambigu, seperti pada pengujian citra di luar dataset dimana ekspresi *disgust* memiliki kemiripan juga dengan ekspresi *sad*.

UCAPAN TERIMA KASIH

Penulis mengucapkan rasa syukur kepada Tuhan Yang Maha Esa karena dengan izin dan berkat-Nya penulis dapat menjalani mata kuliah IF4073 Interpretasi dan Pengolahan Citra ini dari awal sampai pembuatan makalah ini. Terima kasih juga diucapkan kepada Bapak Rinaldi Munir sebagai dosen pengampu mata kuliah ini yang telah mengajarkan kepada penulis materi yang diperlukan dalam pembuatan makalah ini.

VIDEO LINK AT YOUTUBE

link : bit.ly/VideoPencitraanHarith

REFERENCES

- [1] Munir, R. "Pengantar Interpretasi dan Pengolahan Citra (Bagian I)". 2022. Retrieved December 17, 2022, from <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/01-Pengantar-Pengolahan-Citra-Bag1-2022.pdf>
- [2] Munir, R. "Pembentukan Citra dan Digitalisasi Citra". 2022. Retrieved December 17, 2022, from <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/03-Pembentukan-Citra-dan-Digitalisasi-Citra-2022.pdf>
- [3] Munir, R. "24-Convolutional Neural Network". 2022. Retrieved December 17, 2022, from <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/24-CNN-2022.pdf>
- [4] MathWorks. "What Is a Convolutional Neural Network? : 3 things you need to know," (n.d.). Retrieved December 17, 2022, from <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>
- [5] MathWorks. "Deep Learning in MATLAB," (n.d.). Retrieved December 18, 2022, from <https://www.mathworks.com/help/deeplearning/ug/deep-learning-in-matlab.html>
- [6] IBM Cloud Education. "What is Artificial Intelligence (AI)?" (n.d.). Retrieved December 19, 2022, from <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- [7] IBM Cloud Education. "What is machine learning?" (n.d.). Retrieved December 19, 2022, from <https://www.ibm.com/cloud/learn/machine-learning>
- [8] Saha, S. "A comprehensive guide to Convolutional Neural Networks-the eli5 way," November 2022. Retrieved December 19, 2022, from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Gambar 3.5. Hasil pengujian, diuji beberapa citra yang tidak berasal dari *dataset*.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2022

A handwritten signature in black ink, appearing to read 'Harith', with a stylized flourish underneath.

Harith Fakhiri Setiawan
13519161