



BAB 4

ANALISIS DAN HASIL

4.1 Implementasi CNN

Proses *training* dan *testing* akan dilakukan pada pada *hardware* dengan spesifikasi sebagai berikut:

Processor: Intel Celeron CPU 1,0 GHz x 2

RAM: 6 GB

Oprating System: Windows 10

Tools: Jupyter Notebook dan Chrome

Programing Language: Python 3.6

4.1.1 Implementasi Pada Program Python

Implementasi pada bahasa program *python* ini terdiri dari beberap tahap yaitu *Preprocesing*, kemudian perancangan arsitektur CNN, *Training* CNN dan *Testing* pada data *image* dan *testing* secara *real time*. Berikut adalah *source code* dari masing-masing proses tersebut:

1. Preprocessing: Proses yang terjadi pada bagian ini adalah grayscaleing, re-size data, digitalisasi data dan normalisasi. proses tersebut dapat dilihat pada Gambar 4.1 berikut ini.

```
#PREPROCESSING
#Digitalisasi Data
img=cv2.imread("dataset-2/train/mask/mask (1).png')
#Mengubah warna gambar menjadi hitam-putih
gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
#Mengubah ukuran gambar menjadi 100x100
resized=cv2.resize(gray,(100,100))
#Normalisasi data
data=np.array(resized)/255.0
```

Gambar 4.1. Source Code Preprocessing

Arsitektur CNN: Pada tahap ini arsitektur CNN akan dirancang, adapun arsitektur pada penelitian ini terlihat pada Gambar 4.2 berikut.

UIN SUSKA RIAU



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

```
#ARSITEKTUR CNN
#Konvolusi 1 + padding 32 filter dengan Relu dan MaxPooling
model=Sequential()
model.add(Conv2D(32,(3,3),padding='same',input_shape=(100,100,1)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

#Konvolusi 2 + padding 64 filter dengan Relu dan MaxPooling
model.add(Conv2D(64,(3,3),padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

#Flatten Layer untuk melakukan stack pada output konvolusi 2
model.add(Flatten())

#Dropout untuk menonaktifkan beberapa jaringan pada neural network
model.add(Dropout(0.5))

#Dense Layer Input ada 128 neurons
model.add(Dense(128,activation='relu'))

#Pada Layer terakhir menghasilkan 2 output kategori
model.add(Dense(2,activation='softmax'))
```

Gambar 4.2. Source Code Arsitektur CNN

Training CNN: Pada proses ini arsitektur yang dibangun akan dilatih dengan dataset untuk mendapatkan model CNN. Perosesnya pada Gambar 4.3 dibawah ini.

```
#TRAINING
#Pembagian Data
train_data,test_data,train_target,test_target=train_test_split(data,target,test_size=0.1)

#Model Compile
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.001),metrics=['accuracy'])

#Model Fitting
model.fit(train_data, train_target, epochs=20, batch_size=32, validation_split=0.2)
```

Gambar 4.3. Source Code Training

Testing: Model yang dihasilkan akan diuji pada tahap ini, adapapun pengujian pada data image dapat dilihat pada Gambar 4.4 dan pengujian secara real time pada Gambar 4.5 dan Gambar 4.6 dibawah ini.

- (a) Testing Pada Data Image



```
#Load model CNN
model = tf.keras.models.load_model("model-020.model")

# Mendefinisikan Label
categories = ["Tidak menggunakan masker", "Menggunakan masker"]

#Load gambar input
data_test = 'data-test/test (1).jpg'

#Menampilkan gambar input
im = cv2.imread(data_test, cv2.IMREAD_GRAYSCALE)
img = cv2.resize(im, (150, 150))
plt.imshow(img)

#Preprocessing gambar
def prepare(filepath):
    img_size = 150
    img_array = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
    new_array = cv2.resize(img_array, (img_size, img_size))
    return new_array.reshape(-1, img_size, img_size, 1)

#Prediksi
prediction = model.predict([prepare(data_test)])
print("\n", categories[int(prediction[0][0])])
```

Gambar 4.4. Source Code Training Pada Data Image

(b) Testing Pada Data Video

```
#Load Model CNN
model = load_model('model-020.model')

#Load Haar Cascade
face_clsfr=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

#Input Kamera
source=cv2.VideoCapture(0)

#Definisikan Label
labels_dict={0: 'Menggunakan Masker',1: 'Tidak Menggunakan Masker'}
color_dict={0:(0,255,0),1:(0,0,255)}

while(True):
    #Baca input kamera
    ret,img=source.read()
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    #Deteksi wajah
    faces=face_clsfr.detectMultiScale(gray,1.3,5)

    for x,y,w,h in faces:
        #Resize dan normalisasi
        face_img=gray[y:y+w,x:x+w]
        resized=cv2.resize(face_img, (150,150))
        normalized=resized/255.0
        reshaped=np.reshape(normalized, (1,150,150,1))

        #Prediksi
        result=model.predict(reshaped)
        label=np.argmax(result,axis=1)[0]

        #Membuat marka pada wajah
        cv2.rectangle(img,(x,y),(x+w,y+h),color_dict[label],2)
        cv2.rectangle(img,(x,y-40),(x+w,y),color_dict[label],-1)
        cv2.putText(img, labels_dict[label], (x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)

        #Mengeuarkan peringatan jika tidak menggunakan masker
        if label==1:
            os.system('nomask.mp3')
            time.sleep(3)
```

Gambar 4.6. Source Code Training Pada Data Video

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4.1. Hasil Training CNN

Dari training yang dilakukan dengan 1.000 data dengan beberapa kali percobaan. Mekanisme pengujian ini adalah dengan membandingkan beberapa parameter pada arsitektur CNN seperti jumlah Epoch(jumlah iterasi training), nilai



validation split dan *learning rate* (kecepatan pembelajaran). Untuk hasilnya sebagai berikut:

4.1.2.1 Training Berdasarkan Jumlah Epoch

Adapun pada skenario training ini sebagai berikut: gambar *input* = 100x100, *validation split* = 90:10, *learning rate* = 0,001 dan jumlah *epoch* yang dibandingkan adalah 5, 10, dan 20. Adapun hasilnya dapat dilihat pada Tabel ?? berikut ini:

Tabel 4.1

Table 4.1 continued from previous page

No	Epoch	Validation Loss	Training Accuracy	Training Time(s)
1.	5	0,1701	0,9777	900
2.	10	0,1055	0,9805	1742
3.	20	0,0824	0,9986	3493

Dari pengujian tersebut didapat epoch 20 mendapatkan hasil terbaik yaitu dengan nilai *loss* terkecil yaitu 0,0824 dan akurasi tertinggi 0,9986. Namun semakin besar jumlah epoch akan berpengaruh kepada lamanya waktu training (*training time*).

4.1.2.2 Training Berdasarkan Validation Split

Adapun pada skenario training ini sebagai berikut: gambar *input* = 100x100, *epoch* = 20, *learning rate* = 0,001 dan *validation split* yang dibandingkan adalah 70:30, 80:10, dan 90:10. Adapun hasilnya dapat dilihat pada Tabel ?? berikut ini.

Tabel 4.2

Table 4.2 continued from previous page

No	Validation Split	Validation Loss	Training Accuracy	Training Time(s)
1.	70:30	0,1433	0,9982	2433
2.	80:20	0,1055	0,9939	2812
3.	90:10	0,0824	0,9951	3123

Dari hasil pengujian tersebut mendapatkan hasil yang hampir sama pada nilai akurasi tapi yang terlihat perbedaan adalah pada nilai *loss* dan *training time*, semakin besar jumlah data training membuat waktu pelatihan semakin lama tapi menghasilkan nilai *loss* yang lebih kecil.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4.1.2.3 Training Berdasarkan Learning Rate

Adapun pada skenario training ini sebagai berikut: gambar *input* = 100x100, *epoch* = 20, *validation split* = 90:10 dan learning rate yang akan dibandingkan adalah = 0,1, 0,01, dan 0,001. Adapun hasilnya dapat dilihat pada Tabel ?? berikut ini.

No	Learning Rate	Validation Loss	Training Accuracy	Training Time(s)
1	0.1	0,6926	0,4993	900
2	0,01	0,1181	0,9958	1742
3	0,001	0,0824	0,9951	3493

Pada pengujian ini terlihat perubahan nilai *learning rate* sangat berpengaruh pada tahap *training*. Semakin kecil nilai learning rate meningkatkan akurasi dan mengurangi nilai *loss*, namun dengan waktu *training* yang makin lama.

4.1.2.4 Training Berdasarkan Ukuran Gambar

Adapun pada skenario training ini sebagai berikut: *epoch* = 20, *validation split* = 90:10 dan ukuran gambar yang akan dibandingkan adalah = 50x50, 100x100, dan 150x150. Adapun hasilnya dapat dilihat pada Tabel ?? berikut ini.

Tabel 4.4

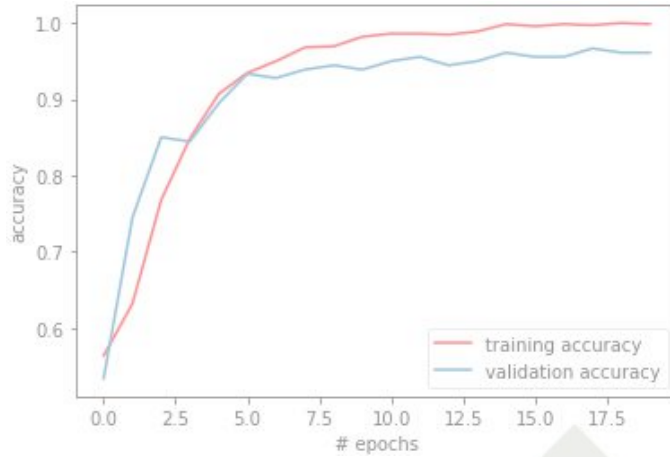
Table 4.4 continued from previous page

No	Image Size	Validation Loss	Training Accuracy	Training Time(s)
1.	50x50	0,1236	0,9026	379
2.	100x100	0,0824	0,9951	3493
3.	150x150	0,0646	0,9986	8130

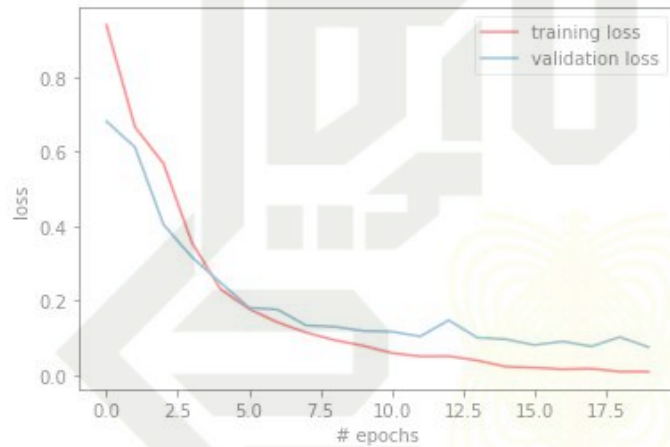
Pada pengujian ini terlihat perubahan ukuran gambar juga berpengaruh pada tahap *training*. Semakin besar ukuran gambar akan meningkat akurasi dan mengurangi nilai *loss*, tapi dengan waktu training lebih makin lama. Dari ketiga skenario pengujian ini maka didapat 1 model terbaik yaitu model dengan kombinasi *epoch* 20, *validation split* 90:10, *learning rate* 0,001 dan input gambar 150x150. Adapun grafik proses *training* model ini dapat digambarkan pada Gambar 4.7 dan Gambar 4.8 berikut ini.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.7. Hasil Prediksi Menggunakan Masker



Gambar 4.8. Hasil Prediksi Tidak Menggunakan Masker

4.1.3 Hasil Testing Model

Dari proses training telah mendapatkan satu model CNN terbaik, selanjutnya model tersebut akan di uji untuk mengetahui kinerja dari model ini. Pengujian ini nantinya akan dikaukan pada 2 kondisi yaitu pengujian pada data Gambar (image) dan pada data video (Real Time).

Dari proses testing pada 100 gambar, model dapat melakukan klasifikasi dengan benar pada seluruh data wajah yang menggunakan masker, sedangkan pada data wajah tanpa menggunakan masker menghasilkan nilai benar pada 49 gambar dan salah pada 1 gambar. Jika dihitung menggunakan rumus akurasi memperoleh hasil 99%. Hasil pengujian dapat dilihat pada confusion matrix di Gambar 4.9 berikut:

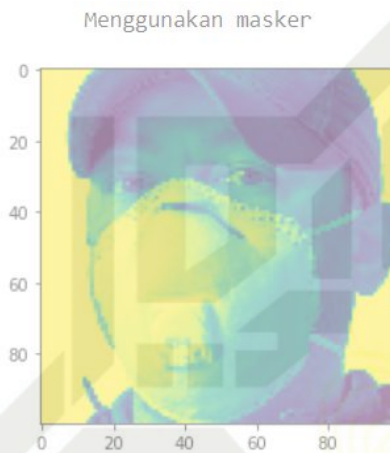
Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

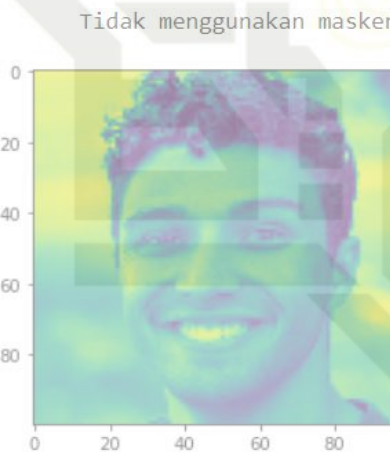
		Predict Class	
		Mask	No Mask
Actual Class	Mask	49	1
	No Mask	0	50

Gambar 4.9. Hasil Prediksi Tidak Menggunakan Masker

Kemudian Contoh hasil prediksi pada data gambar dapat dilihat pada Gambar 4.10 dan Gambar 4.11 berikut ini.



Gambar 4.10. Hasil Prediksi Menggunakan Masker



Gambar 4.11. Hasil Prediksi Tidak Menggunakan Masker

Dari Hasil pengujian secara *real time* pada Toko Buku Zanafra, yang mana pengujian dilakukan pada 20 orang pengunjung toko yaitu disini aplikasi ini berhasil mendeteksi dengan benar pada 19 orang dan hanya terdapat satu yang miss klasi-

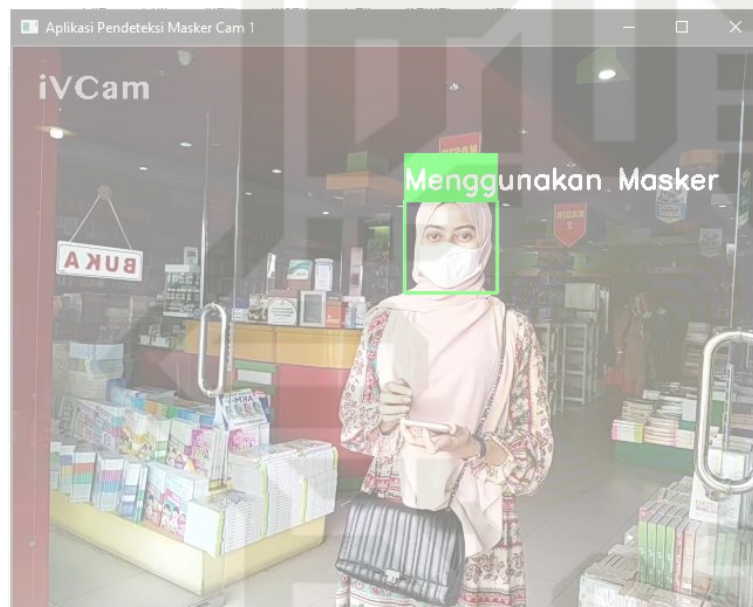


fikasi. Dengan akurasi 95% Hasil pengujian dapat dilihat pada *confution matrix* pada Gambar 4.12 berikut:

		Predict Class	
		Mask	No Mask
Actual Class	Mask	9	1
	No Mask	0	10

Gambar 4.12. Hasil Prediksi Tidak Menggunakan Masker

Gambar 4.13 dan Gambar 4.14 adalah contoh hasil pengujian ini yang tealh dilakukan pada toko buku Zanafa.



Gambar 4.13. Hasil Prediksi Menggunakan Masker

- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.14. Hasil Prediksi Tidak Menggunakan Masker

© Hak cipta milik UIN Suska Riau

State Islamic University of Sultan Syarif Kasim

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masa
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.