DAFTAR PUSTAKA

[1] M. Rizky Firdaus, "Penerapan Algoritma Convolutional Neural Network dalam Klasifikasi Telur Ayam Fertil dan Infertil Berdasarkan Hasil Candling," *Jurnal Informatika Universitas Pamulang,* Vol. 5, No.4, 2020.

[2] Shoffan Saifullah dan A. P. Suryotomo, "Identification if Chicken Egg Fertility Using SVM Classifier based on Firs-Order Statistical Feature Extraction," *Jurnal ilmiah,* vol. Vol.13 No.3, 2021.

[3] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks," towards data science, 16 Desember 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Diakses 15 Desember 2021].

[4] Admin, "Permasalahan yang Kerap Timbul Saat Penetesan Telur Itik," Pertanianku.com, 27 Januari 2020. [Online]. Available: https://www.pertanianku.com/permasalahan-yang-kerap-timbul-saat-penetasan-telur-itik/. [Diakses 15 Desember 2021].

[5] Admin, "Digital Image Processing," University of Tartu, Januari 2014. [Online]. Available: https://sisu.ut.ee/imageprocessing/book/1. [Diakses 16 Desember 2021].

[6] Admin, "Deep Learning," IBM, 01 Mei 2020. [Online]. Available: https://www.ibm.com/cloud/learn/deep-learning. [Diakses 16 Desember 2021].

[7] Admin, "What is Deep Learning and How does it work ?," Towards Datascience, [Online]. Available: https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac. [Diakses 16 Desember 2021].

[8] S. Sena, "Pengenalan Deep Learning Part 7: Convolutional Neural Network," Medium, 2017. [Online]. Available: https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-

convolutional-neural-network-cnn-b003b477dc94. [Diakses 17 Desember 2021].

[9] N. Sofia, "Convolutional Neural Network," Medium, [Online]. Available: https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b. [Diakses 17 Desember 2021].

[10] R. Nanculef, P. Radeva dan S. Balocco, "Training Convolutional Nets to Detect Calcified Plaque in IVUS Sequences," *Intravascular Ultrasound,* vol. Chapter 9, pp. 141-158, 2020.

[11] K. Smeda, "Understand the Architecture of CNN," Towards Datascience, 01 November 2019. [Online]. Available: https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7. [Diakses 25 Desember 2021].

[12] S. Albelwi dan A. Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks," *Entropy,* 2017.

[13] A. Razor, "Push Button Arduino: Pengertian, Fungsi, dan Prinsip Kerja," Aldyrazor.com, 2017. [Online]. Available: https://www.aldyrazor.com/2020/05/push-button-arduino.html. [Diakses 19 Desember 2021].

[14] Admin, "What is a Raspberry Pi?," opensource.com, [Online]. Available: https://opensource.com/resources/raspberry-pi. [Diakses 17 Desember 2021].

[15] Admin, "About," Opencv, [Online]. Available: https://opencv.org/about/. [Diakses 20 Desember 2021].

[16] Admin, "What is a Buzzer : Working & Its Application," EL-PRO-CUS, [Online]. Available: https://www.elprocus.com/buzzer-working-applications/. [Diakses 20 Desember 2021].

[17] D. R. Salsabila, "Sistem Pendeteksi Gejala Awal Tantrum pada Anak Autisme melalui Ekspresi Wajah dengan Convolutional Neural Network," Universites Andalas, Padang, 2021.

[18] D. Purwaningsih, M. A. Djaelani dan T. R. Saraswati, "Kualitas Telur Ayam Ras Setelah Pemberian Olesan Lidah Buaya (Aloe vera) dan Lama," *Buletin Anatomi dan Fisiologi,* Vol. %1 dari %2volume 24, Nomor 1, pp. 13-20, 2016.

[19] A. Nurul, "jiptummpp-gdl-azizahnurl-51531-3-babii.pdf," [Online]. Available: https://eprints.umm.ac.id/36840/3/jiptummpp-gdl-azizahnurl-51531-3-babii.pdf. [Diakses 06 Maret 2022].

[20] A. Pratama, "Prototipe Ayunan Bayi Otomatis Berbasis Mikrokontroler dengan Notifkasi Telegram," Universitas Andalas, Padang, 2021.

[21] K. S. Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning," Medium, 13 November 2019. [Online]. Available: https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f. [Diakses 07 Maret 2022].

[22] N. Ashikin, "Alat Pengecek Telur Fertil atau Infertil Menggunakan Kamera," Institut Teknologi Sepuluh November, Surabaya, 2018.

[23] J. G. C, Rancapan, E. R.Arboleda, J. L, D. Jr dan R. M. Dellosa, "Egg Fertility Detection Using Image Processing and Fuzzy Logic," *International Journal of Scientific & Technology Research,* vol. Vol. 8, pp. 3228-3230, 2019.

[24] M. S. Anggreany, "School of Computer Science," Binus University, 01 November 2020. [Online]. Available: https://socs.binus.ac.id/2020/11/01/confusion-matrix/#:~:text=Confusion%20Matrix%20adalah%20pengukuran%20performa,nilai%20prediksi%20dan%20nilai%20aktual.. [Diakses 06 Maret 2022].

[25] M. R. P. Anggara, "Transfer Learning Rate dengan Model Arsitektur VGG16 dan Inception-V3 untuk Klasifikasi Penyakit Daun Kentang," Institut Teknologi Nasional, Bandung, 2020.

[26] Q. Lina, "Apa itu Convolutional Neural Network ?," Medium, 2 Januari 2019. [Online]. Available: https://medium.com/@16611110/apa-itu-

convolutional-neural-network-836f70b193a4. [Diakses 08 Maret 2022].

[27] Admin, https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/, Diakses 20 Juni 2023

LAMPIRAN

Lampiran 1 : Source Code

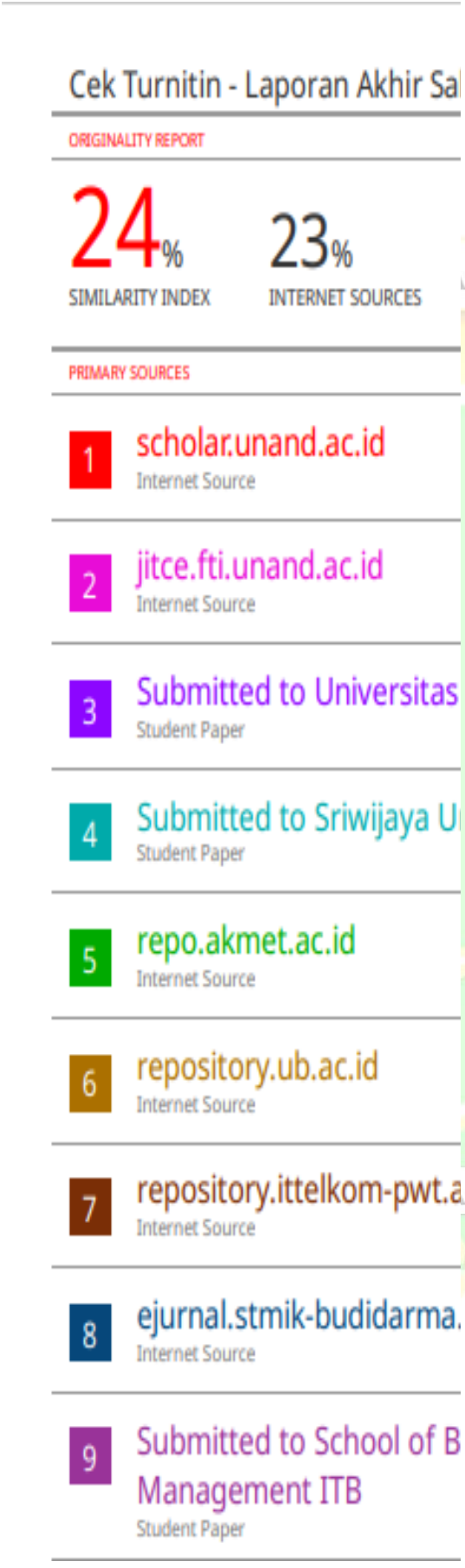
```
from dotenv import
dotenv_values

# imports telegram bot
from telegram import Bot
from telegram import
Update, ReplyKeyboardMarkup
from telegram.ext import
Updater, CommandHandler,
MessageHandler, Filters,
CallbackContext
from telegram import
InputFile

import tensorflow as tf
import cv2
import numpy as np
from time import sleep,
time
import RPi.GPIO as GPIO
from tcs3200 import TCS3200
from threading import
Thread

config =
dotenv_values(".env")

TELEGRAM_TOKEN =
config["TELEGRAM_TOKEN"]

updater =
Updater(token=TELEGRAM_TOKE
N, use_context=True)
dispatcher =
updater.dispatcher
user_states = {}

# load model
with open ('model.json',
'r') as json_file:
    json_saved_Model =
json_file.read()

# initialize model
model_j =
tf.keras.models.model_from_
json(json_saved_Model)
model_j.load_weights('model
.h5')


model_j.compile(

loss='sparse_categorical_cr
ossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

# GPIO pins definition
BUTTON_PIN = 18
RELAY_PIN = 14

LED_RED_PIN = 13
LED_GREEN_PIN = 19
BUZZER_PIN = 6


# setup GPIOs
GPIO.setwarnings(False) #
ignore warnings
GPIO.setmode(GPIO.BCM) #
set GPIO mode to BCM

GPIO.setup(BUTTON_PIN,
GPIO.IN,
pull_up_down=GPIO.PUD_UP)
GPIO.setup(LED_GREEN_PIN,GP
IO.OUT)
GPIO.setup(LED_RED_PIN,GPIO
.OUT)
GPIO.setup(BUZZER_PIN,GPIO.
OUT)
GPIO.setup(RELAY_PIN,
GPIO.OUT)

color_sensor =
TCS3200(oe_pin = 8, s2_pin
= 23, s3_pin = 24,
signal_pin = 25)

def start(update:Update,
context: CallbackContext) -
> None:
    keyboard = [
        ["Rekap klasifikasi
data"]
    ]
```

```
    reply_markup =
ReplyKeyboardMarkup(keyboar
d, one_time_keyboard=True)


update.message.reply_text(
        "Hello! Pilihlah
opsi dibawah ini untuk
melihat klasfikasi data",

reply_markup=reply_markup
    )


def
rekap_klasifikasi_data(upda
te, context:
CallbackContext) -> None:
    chat_id =
update.message.chat_id
    file_path =
'classification_output.txt'

    with open(file_path,
'rb') as file:

context.bot.send_document(c
hat_id=chat_id,
document=file)


user_states[update.message.
from_user.id] = None

def
handle_user_input(update:
Update, context:
CallbackContext) -> None:
    user_id =
update.message.from_user.id
    user_input =
update.message.text

    if user_input ==
"/start":

user_states[user_id] = None
        start(update,
context)
    else:

user_states[user_id] =
user_input
```

```
start_handler =
CommandHandler('start',
start)
send_file_handler =
MessageHandler(Filters.text
("Rekap klasifikasi data"),
rekap_klasifikasi_data)
user_input_handler=
MessageHandler(Filters.text
& ~Filters.command,
handle_user_input)

dispatcher.add_handler(star
t_handler)
dispatcher.add_handler(send
_file_handler)
dispatcher.add_handler(user
_input_handler)

def grab_image(post_delay =
2):
    print("[INFO]\t--
Opening camera..")
    cap =
cv2.VideoCapture(0)
    print("[INFO]\t--
Capturing single frame..")
    ret, frame = cap.read()

    sleep(1)

    print("[INFO]\t--
Releasing camera..")
    cap.release()

    print(f"[INFO]\t--Post
Delaying.. ({time()})")
    sleep(post_delay)
    print(f"[INFO]\t--Done
({time()})")

    return frame

def in_range(val, val_from,
val_to):
    return (val >=
val_from) and (val <=
val_to)

def beep_buzzer():
```

```
    print("[INFO]\tBeeping
buzzer..")
    GPIO.output(BUZZER_PIN,
GPIO.HIGH)
    sleep(0.5)
    GPIO.output(BUZZER_PIN,
GPIO.LOW)

def turn_lamp_on():
    print("[INFO]\tLamp
on")
    GPIO.output(RELAY_PIN,
GPIO.HIGH)

def turn_lamp_off():
    print("[INFO]\tLamp
off")
    GPIO.output(RELAY_PIN,
GPIO.LOW)

def
is_egg_fertil(egg_count,
save_image = True):
    # grab image
    turn_lamp_on()
    img = grab_image()
    turn_lamp_off()

    if save_image:

print("[INFO]\tImage has
been saved")

cv2.imwrite(f"images/captur
ed_image_{egg_count}.jpg",
img)

    # pre-process image
    IMG_SIZE = 224
    rgb_img =
cv2.cvtColor(img,
cv2.COLOR_BGR2RGB)
    rgb_img =
cv2.resize(rgb_img,
(IMG_SIZE,IMG_SIZE))
    rgb_img =
np.resize(rgb_img,[1,IMG_SI
ZE,IMG_SIZE,3])

    # predict fertility
    predict_x =
model_j.predict(rgb_img)
    classes_x =
np.argmax(predict_x,axis=1)

    is_fertil = (classes_x
== 0)

    return is_fertil

def main_task():
    print("[INFO]\t Main
Task Started.")

    # load egg count from
previous reading
    with
open("output_count.txt",
"r") as count_file:
        egg_count =
int(count_file.read().strip
())

    while True:

        # read current
button state
        button_state =
GPIO.input(BUTTON_PIN)

        if button_state ==
True: continue

        # button pressed
        print("[INFO]\t
Button pressed.")

        is_fertil =
is_egg_fertil(egg_count)

        if is_fertil:

print("[INFO]\tTelur
fertil")
            beep_buzzer()

GPIO.output(LED_RED_PIN,
GPIO.LOW)

GPIO.output(LED_GREEN_PIN,
GPIO.HIGH)
        else:

print("[INFO]\tTelur
infertil")
```

```
GPIO.output(LED_RED_PIN,
GPIO.HIGH)

GPIO.output(LED_GREEN_PIN,
GPIO.LOW)

        # classifiying egg
type
        max_retry = 5
        for i in
range(max_retry):
            r, g, b =
color_sensor.get_rgb()

            egg_type =
"UNKNOWN"
            if in_range(r,
13000, 15000) and
in_range(g, 23629, 25055)
and in_range(b, 21119,
25085):

print("[INFO]\tTelur ayam
ras")
                egg_type =
"Telur ayam ras"
            elif
in_range(r, 19039, 20733)
and in_range(g, 20570,
29725) and in_range(b,
25637, 28591):

print("[INFO]\tTelur ayam
kampung")
                egg_type =
"Telur ayam kampung"
            elif
in_range(r, 17000, 29167)
and in_range(g, 25221,
27758) and in_range(b,
23224, 28513):

print("[INFO]\tTelur
bebek")
                egg_type =
"Telur bebek"
            else:

print("[INFO]\tTelur tidak
dikenali")
            if egg_type !=
"UNKNOWN":
                break


print(f"[INFO]\tRetrying
({i + 1})..")
            egg_count += 1
            with
open("output_count.txt",
"w") as count_file:

count_file.write(str(egg_co
unt))

            output_text =
f"Classification
{egg_count} Result:
{egg_type}, {'FERTIL' if
is_fertil else 'INFERTIL'}
\n"
            with
open(f"classification_outpu
t.txt", "a") as
output_file:

output_file.write(output_te
xt)


thread =
Thread(target=main_task)
thread.start()

updater.start_polling()
updater.idle()
```



Cek Turnitin - Laporan Akhir Sa

ORIGINALITY REPORT

24% SIMILARITY INDEX

23% INTERNET SOURCES

PRIMARY SOURCES

1 scholar.unand.ac.id
Internet Source

2 jitce.fti.unand.ac.id
Internet Source

3 Submitted to Universitas
Student Paper

4 Submitted to Sriwijaya U
Student Paper

5 repo.akmet.ac.id
Internet Source

6 repository.ub.ac.id
Internet Source

7 repository.ittelkom-pwt.a
Internet Source

8 ejurnal.stmik-budidarma.
Internet Source

9 Submitted to School of B Management ITB
Student Paper