LAMPIRAN

Program Merekam Menggunakan Raspberry Pi dan Pi Camera

```
 import numpy as npimport os
import cv2
 filename = 'video.avi' frames_per_second = 24.0res = '720p'
 # Set resolution for the video capture
 # Function adapted from https://kirr.co/0l6qmhdef change_res(cap,
 width, height):
 cap.set(3, width) cap.set(4, height)
 # Standard Video Dimensions Sizes STD_DIMENSIONS =  {
 "480p": (640, 480),
720p": (1280, 720),
 "1080p": (1920, 1080),
 "4k": (3840, 2160),
}

 # grab resolution dimensions and set video capture to it.def
 get_dims(cap, res='1080p'):
 width, height = STD_DIMENSIONS["720p"]if res in STD_DIMENSIONS:
 width,height = STD_DIMENSIONS[res]## change the current caputre
 device ## to the resulting resolution change_res(cap, width,
 height)
 return width, height
  # Video Encoding, might require additional installs# Types of
  Codes: http://www.fourcc.org/codecs.php VIDEO_TYPE = {
 'avi': cv2.VideoWriter_fourcc(*'XVID'), #'mp4':
 cv2.VideoWriter_fourcc(*'H264'),'mp4':
 cv2.VideoWriter_fourcc(*'XVID'),
 }
 def get_video_type(filename):
 filename,ext = os.path.splitext(filename)if ext in VIDEO_TYPE:
 return VIDEO_TYPE[ext]return VIDEO_TYPE['avi']
 cap = cv2.VideoCapture(0)
 out = cv2.VideoWriter(filename, get_video_type(filename), 25,
 get_dims(cap, res))

 while True:
 ret, frame = cap.read() out.write(frame) cv2.imshow('frame',frame)
 if cv2.waitKey(1) & 0xFF == ord('q'):break
 cap.release() out.release() cv2.destroyAllWindows()
```

Program Konversi Video Ke Gambar

```
Import cv2
import os
def video_to_frames(video, path_output_dir): vidcap =
cv2.VideoCapture(video)
count = 0
while vidcap.isOpened():
success, image = vidcap.read() if cv2.waitKey(1)& 0xFF == 32:
cv2.imwrite(os.path.join(path_output_dir, 'kendaraan%d.png') %
count, image)
count += 1 else:
break cv2.destroyAllWindows() vidcap.release()
```

Normalisasi Gambar

```
from PIL import Image import os

import argparse


def rescale_images(directory, size): for img in
os.listdir(directory):

im = Image.open(directory+img)

im_resized = im.resize(size, Image.ANTIALIAS)
im_resized.save(directory+img)


if name == ' main ':

parser = rgparse.ArgumentParser(description="Rescaleimages")
parser.add_argument('-d', '--directory', type=str, required=True,
help='Directory containing the images') parser.add_argument('-s',
'--size', type=int, nargs=2, required=True, metavar=('width',
'height'), help='Image size')

args = parser.parse_args() rescale_images(args.directory,
args.size)
```

Program deteksi Manusia

```
import numpy as np
import math
import time
import sys
import cv2
import argparse
import datetime
import telepot

from tensorflow.lite.python.interpreter import Interpreter

fps = ""
detectfps = ""
framecount = 0
detectframecount = 0
time1 = 0
time2 = 0
EntranceCounter = 0
EntranceCounter2 = 0

def CheckEntranceLineCrossing(ymax, CoorXEntranceLine):
    AbsDistance = abs(ymax - CoorXEntranceLine)

    if AbsDistance <= 9:
        return 1
    else:
        return 0

def CheckEntranceLineCrossing2(ymax, CoorXEntranceLine2):
    AbsDistance = abs(ymax - CoorXEntranceLine2)

    if AbsDistance <= 9:
        return 1
    else:
        return 0

api = '6139415220:AAHESYxYrE5O0-I9QXYtd08RuxxjgdoOGg' #token dari
bot telegram
bot = telepot.Bot(api)

LABELS = ['person','Aan','Mamah', 'Edo']

if __name__ == '__main__':

    parser = argparse.ArgumentParser()
    parser.add_argument("--model",           default="ssd.tflite",
help="Path of the detection model.")
    parser.add_argument("--num_threads",    type=int,    default=9,
help="Threads.")
    args = parser.parse_args()

    model        = args.model
    num_threads  = args.num_threads

    interpreter = Interpreter(model_path=model)
    interpreter.set_num_threads(num_threads)
```

```
    interpreter.allocate_tensors()
    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()

    cam = cv2.VideoCapture(0)

    window_name = "Movie"
    cv2.namedWindow(window_name, cv2.WINDOW_AUTOSIZE)


    while True:
        start_time = time.perf_counter()

        ret, image = cam.read()

        # Resize and normalize image for network input
        image_height = image.shape[0]
        image_width = image.shape[1]
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame = np.expand_dims(frame, axis=0)
        frame = frame.astype(np.float32)
        cv2.normalize(frame, frame, -1, 1, cv2.NORM_MINMAX)
        cv2.putText(image,   fps,   (image_width   -   170,   15),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (38, 0, 255), 1, cv2.LINE_AA)
        cv2.putText(image,  detectfps,  (image_width  -  170,  30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (38, 0, 255), 1, cv2.LINE_AA)

        # get results
        boxes                                                    =
interpreter.get_tensor(output_details[0]['index'])[0]
        classes                                                  =
interpreter.get_tensor(output_details[1]['index'])[0]
        scores                                                   =
interpreter.get_tensor(output_details[2]['index'])[0]
        count                                                    =
interpreter.get_tensor(output_details[3]['index'])[0]

        CoorXEntranceLine = int(image_width)
        cv2.line(image,     (int(CoorXEntranceLine),     0     ),
(int(CoorXEntranceLine), image_height), (255, 0, 100), 5)

        # draw boxes
        for  i,  (box,  classidx,  score)  in  enumerate(zip(boxes,
classes, scores)):
            probability = score

            if probability >= 0.45:
                if not box[0] or not box[1] or not box[2] or not
box[3]:
                    continue
                ymin = int(box[0] * image_height)
                xmin = int(box[1] * image_width)
                ymax = int(box[2] * image_height)
                xmax = int(box[3] * image_width)

                classnum = int(classidx)
                #print('coordinates:  ({},  {})-({},  {}).  class:
"{}". probability: {:.2f}'.format(xmin, ymin, xmax, ymax, classnum,
```

```
score))
                cv2.rectangle(image, (xmin, ymin), (xmax, ymax),
(0, 255, 0), 2)
                cv2.putText(image,                              '{}:
{:.2f}'.format(LABELS[classnum],  score),  (xmin,  ymin  -  5),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0), 2)
                #ini tambahan
                cv2.imwrite('orang.jpg', image)

                date  =  datetime.datetime.now().strftime("%Y-%m-
%d")
                jam = datetime.datetime.now().strftime("%H:%M:%S")

                CoordXCentroid = int((xmax))
                cv2.circle(image,  ObjectCentroid,  1,  (255  ,255
,255), 10)

                if(CheckEntranceLineCrossing(CoordXCentroid,
CoorXEntranceLine)):

                    EntranceCounter += 1
                    if classnum == 0:
                        print('Person')

bot.sendPhoto("6269595626",photo=open("orang.jpg","rb"))    #untuk
mengirim gambar ke telegram #gianti
                        bot.sendMessage("6269595626",    "Person")
#untuk memberikan notif ke telegram
                        print('Person' ,date, jam)

                    elif classnum == 1:
                        print('Aan',date, jam)

bot.sendPhoto("6269595626",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Aan")

                    elif classnum == 2:
                        print('Mamah' ,date, jam)

bot.sendPhoto("6269595626",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Mamah")

                    elif classnum == 3:
                        print('Edo',date, jam)

bot.sendPhoto("6269595626",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Edo")

                    else:
                        print('Tidak terdeteksi')

                if(CheckEntranceLineCrossing2(CoordXCentroid,
CoorXEntranceLine2)):

                    EntranceCounter2 += 1
                    if classnum == 0:
                        print('Person')

bot.sendPhoto("6269595626",photo=open("orang.jpg","rb"))    #untuk
```

```
mengirim gambar ke telegram #gianti
                        bot.sendMessage("6269595626",    "Person")
#untuk memberikan notif ke telegram
                        print('Person' ,date, jam)

                    elif classnum == 1:
                        print('Aan',date, jam)

bot.sendPhoto("6269595690",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Aan")

                    elif classnum == 2:
                        print('Mamah' ,date, jam)

bot.sendPhoto("6269595690",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Mamah")

                    elif classnum == 3:
                        print('Edo',date, jam)

bot.sendPhoto("6269595690",photo=open("orang.jpg","rb"))
                        bot.sendMessage("6269595626", "Edo")

                    else:
                        print('Tidak terdeteksi')

        cow = cv2.resize(image, (720,480))
        cv2.imshow(window_name, cow)


        if cv2.waitKey(1)&0xFF == ord('q'):
            break

        detectframecount += 1
        # FPS calculation
        framecount += 1
        if framecount >= 10:
            fps = "(Playback) {:.1f} FPS".format(time1 / 10)
            detectfps         =         "(Detection)         {:.1f}
FPS".format(detectframecount / time2)
            framecount = 0
            detectframecount = 0
            time1 = 0
            time2 = 0
        end_time = time.perf_counter()
        elapsedTime = end_time - start_time
        time1 += 1 / elapsedTime
        time2 += elapsedTime
```