

```
1  from gpiozero import MotionSensor
2  import time
3  import picamera
4  from time import sleep
5  from datetime import datetime
6  import os
7  from moviepy.editor import VideoFileClip
8  import numpy as np
9  from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input,
decode_predictions
10
11  folder_path = 'home/klaule/Videos/'
12
13  if not os.path.exists(folder_path):
14      os.makedirs(folder_path)
15
16  pir = MotionSensor(23, threshold=0.5)
17
18  #laden des vortrainierten MobileNetV2-Modells
19  model = MobileNetV2(weights='imagenet')
20
21  def classify_image(image):
22      #bild für MobileNetV2 vorverarbeiten.
23      image = cv2.resize(image,(224, 224))
24      image = preprocess_input(image)
25      image = np.expand_dims(image,axis=0)
26
27      #klassifizierung durchführen
28      predictions = model.predict(image)
29
30      #tierklassen-id aus ImageNet
31      animal_class_ids = [16, 22, 27, 31, 32, 33, 49,50, 51]
32
33      #überprüfen, ob eine Tierklasse in den top-3 vorhersagen vorhanden ist.
34      top_predictions = decode_predictions(predictions, top3)[0]
35      for _, _, class_id in top_predictions:
36          if class_id in animal_class_ids:
37              return True
38
39      return False
40
41  def main():
42      with picamera,PiCamera() as camera:
43          while True:
44              pir.wait_for_motion()
45              print("Bewegung erkannt")
46
47              #generiere dateiname basierend auf dem aktuellen datum und uhrzeit
48              date_string = datetime.now().strftime("%Y%m%d_%H%M%S")
49              filename = f"{date_string}.h264"
50
51
```

```
52         #vorbereitung der kamera
53         camera.resolution = (640, 480)
54         camera.framerate = 30
55         #kamera um 180 drehen
56         camera.rotation = 180
57
58         #aufnahme starten
59         camera.start_preview()
60         camera.start_recording(os.path.join(folder_path, filename))
61
62         #video aufnahme 5s
63         sleep(5)
64
65         #aufnahme stoppen
66         camera.stop_recording()
67
68         #videodatei in OpenCV-Format konvertieren
69         video_capture = cv2.VideoCapture(filename)
70
71         #frames durchgehen und tiererkennung durchführen
72         while True:
73             ret, frame = video_capture.read()
74             if not ret:
75                 break
76
77             #tiererkennung durchführen
78             if classify_image(frame):
79                 print("Tier erkannt! Speichere Video ab.")
80
81             #code zum speichern des video mit tier
82             video_filename = ("Tier Video" + filename)
83             break
84
85         #aufräumen
86         video_capture.release(folder_path, video_filename)
87         cv2.destroyAllWindows()
88
89 if __name__ == "__main__":
90     main()
91
92
93
```