

Projekt Foto past

Vypracoval Barnabáš Bernášek

2022/2023

Obsah

- | | |
|-----------------|---|
| 1. Úvod | 3 |
| 2. Postup práce | |

Úvod

Mým cílem je vytvořit funkční foto/video past pro Raspberry Pi počítače s využitím Raspberry Pi Camera V2. Zařízení je vhodné pro zachytávání fotek svých mazlíčků, rekreativnímu sledování divých zvířat nebo i pro domácí zabezpečení.

S fotopastí můžete buďto vyfotit fotky nebo nahrávat video, jestli to bude fotka nebo video a jak bude video dlouhé si určuje uživatel pomocí jednoduchých příkazů přes terminál. Jako jméno jsem vybral 'motcam' pro anglickou zkratku z 'motion camera' -> 'motcam'.

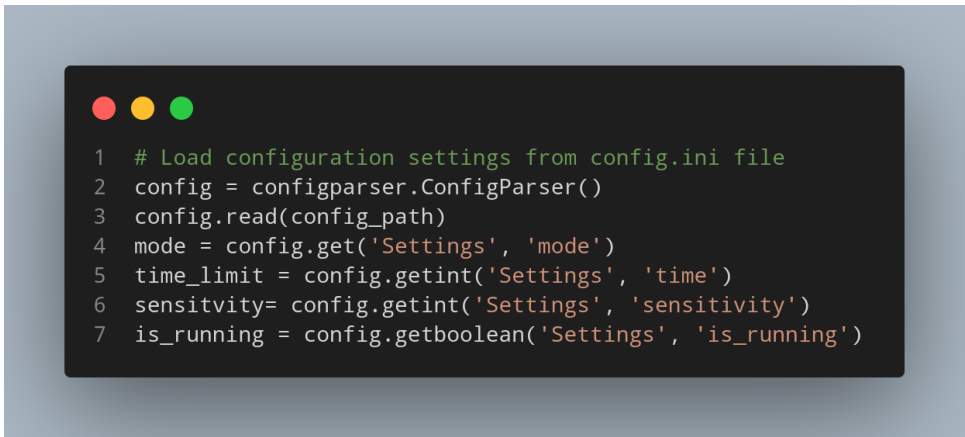
Postup práce

Jako hlavní jazyk jsem si vybral Python 3, kvůli lehkému ovládání kamery s Python knihovnami. Můj skript zejména využívá knihovnu cv2 – OpenCV, která značně zlehčuje práci s kamerou a detekcí pohybu.

Celý projekt se skládá z hlavního Python skriptu(motcam_script.py), který se stará o zachytávání fotek/videí, konfiguračního souboru(config.ini), zde se ukládají informace o nastavení foto pastí, vybral jsem si konfigurační soubor s koncovkou .ini kvůli lehké čitelnosti a zápisu. Soubor motcam se stará o komunikaci s uživatelem přes terminál, zda-li chce uživatel spustit motcam, upravit, nebo zastavit, poslední soubor je set_up.sh, který by se měl po stáhnutí spustit jako první, stará se o nastavení vlastních příkazů(motcam –start, motcam –edit, motcam –stop).


1. Hlavní skript(motcam_script.py)

Hlavní skript se skládá z jednotlivých částí, které teď budu popisovat.




```
1 # Load configuration settings from config.ini file
2 config = configparser.ConfigParser()
3 config.read(config_path)
4 mode = config.get('Settings', 'mode')
5 time_limit = config.getint('Settings', 'time')
6 sensitivity= config.getint('Settings', 'sensitivity')
7 is_running = config.getboolean('Settings', 'is_running')
```

Jako první si načteme hodnoty z config.ini souboru, s těmito hodnotami budeme pracovat později.



```
1 if mode != 'photo' and mode != 'video':
2     print(mode)
3     print("Wrong input at mode")
4     can_start = False
5
6 if time_limit < 0:
7     print("Wrong input at time limit, cannot be 0 or less then 0")
8     can_start = False
9
10 if sensitivity < 1000 or sensitivity <= 0:
11     print("Wrong input at sensitivity")
```

Dále se musíme ujistit zda jsou hodnoty správně zadane uživatelem. Když nejsou, vypíše se kde udělal uživatel chybu.



```
1 if can_start:
2     # Initialize camera and motion detector
3     camera = None
4     detector = cv2.createBackgroundSubtractorMOG2()
```

Zde nastavíme kameru a detektor.

```
1 # Create folder to store captured files if it does not already exist
2 if not os.path.exists('Captures'):
3     os.makedirs('Captures')
4
```

Dále vytvoříme novou složku Captures, kam se budou ukládat naše fotky/video zachycené pastí.

```
1 # Main loop to capture photos or videos when motion is detected
2 while is_running:
3     try:
4         # Initialize camera if it has not already been initialized
5         if camera is None:
6             camera = cv2.VideoCapture(0)
7
8         # Capture a frame from the camera and convert it to grayscale
9         ret, frame = camera.read()
10        if not ret:
11            raise Exception("Error: Camera read failed")
12
13        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
14
15        # Apply background subtraction to the frame to detect motion
16        mask = detector.apply(gray)
17
18        # Check if enough motion has been detected to trigger a capture
19        motion = (cv2.countNonZero(mask) > sensitivity)
20
```

Hlavní cyklus zachycování - vytvoření šedé první fotky podle které se dále zkoumá pohyb.

```
1 # If motion has been detected, capture a photo or video based on the configuration settings
2 if motion:
3     filename = f"motion_captured_on_{datetime.now()}"
4     if mode == 'photo':
5         time.sleep(0.5)
6         cv2.imwrite(f'Captures/{filename}.jpg', frame)
7     elif mode == 'video':
8         time.sleep(0.5)
9         fourcc = cv2.VideoWriter_fourcc(*'mp4v')
10        out = cv2.VideoWriter(f'Captures/{filename}.mp4', fourcc, 20.0, (640, 480))
11        start_time = time.time()
12        while (time.time() - start_time) < time_limit:
13            ret, frame = camera.read()
14            if not ret:
15                raise Exception("Error: Camera read failed")
16            out.write(frame)
17        out.release()
18
```

Dále už jen vyfotíme/nahrajeme pokud jsme zachytili pohyb.



```
1 # Wait for a short period of time before capturing the next frame
2 time.sleep(0.1)
```



```
1 # Check the is_running setting in the config file to see if the loop should be stopped
2 config.read(config_path)
3 is_running = config.getboolean('Settings', 'is_running')
```



```
1 except KeyboardInterrupt:
2     # If the user presses CTRL+C, release the camera and video writer objects
3     print("\nStopping the script...")
4     if camera is not None:
5         camera.release()
6     cv2.destroyAllWindows()
7     break
8
9 except Exception as e:
10    # If any other exception occurs, release the camera and video writer objects and print the error message
11    print(f"Error occurred: {e}")
12    if camera is not None:
13        camera.release()
14    cv2.destroyAllWindows()
15    break
16
```

Dále máme výjimky na nečekané ukončení.



```
1 # Release the camera and video writer objects
2 if camera is not None:
3     camera.release()
4 cv2.destroyAllWindows()
```

A nakonec uvolnění kamery.

2. Webová stránka

Součástí mého projektu je také webová stránka, přístupná na doméně www9.spse.1984.cz/ nebo pomocí QR kódu.

Webová stránka obsahuje fotky mých koček, fotky byli pořízeny přes celý den co jsem se účastnil vyučování ve škole. Tento test proběhl úspěšně - past zachytila přes 1500 fotek. Pár z těchto fotek jsem se rozhodl použít pro tuto webovou stránku, jako ukázkou funkčnosti.