

Applikation

Zur Ausführung den *Application* Ordner mit Folgendem Inhalt auf den Raspberry Pi kopieren:

```
├─ Application/
│   ├── models/
│   │   ├── animals_faster_rcnn_inception/
│   │   │   ├── frozen_inference_graph.xml
│   │   │   └── ...
│   ├── launcher.sh
│   ├── main.sh
│   ├── detection.py
│   └── connection.py
```

main.py ist die Hauptanwendung, welche detection.py und connectoin.py verwendet.

aufgerufen wird das main Script über das launcher.sh Script

Im *models* Ordner sind OpenVino Modelle die auf folgende Datensätze trainiert wurden:

- Animals: 9 Wildtierklassen:
- Samples: Alltagsgegenstände zum testen der Anwendung
 - Klassen sind in *classes.txt* der jeweiligen Modell-Ordner definiert

Diese enthalten jeweils folgende Modelle:

- Faster R-CNN: wird von der Applikation default mäßig verwendet
- SSD: wird verwendet, wenn beim Laden des Faster R-CNN ein Fehler auftritt

Einstellungen

Die Anwendung kann mit oder ohne Senden der Daten ausgeführt werden.

Um die Daten senden zu können wird ein remote.it Account benötigt, wie in Connection/remote_it/readme erklärt.

Dafür müssen folgende Einstellung vorgenommen werden:

```
# Gibt mit True/False ob erkannte Bilder gesendet werden sollen
send_results = True

# User Name des Gerätes, an das die Bilder gesendet werden sollen
remote_user = ''

# Device Name des Gerätes in remote.it
remote_divice_name = ''

# E-Mail Adresse des remot.it Accounts
remote_it_email = ''
```

```
# Rassort des Gerätes, an das die Bilder gesendet werden sollen
password_remote_divece = ''

# Rassort des des remot.it Accounts
password_remoteit = ''

# Pfad im Zielgerät, in dem die Bilder gespeichert werden sollen
remote_output_dir = os.path.join('/home', remote_user)

# (optinal) E-Mail Adresse an die eine benachrichtigung gesendet werden
soll
send_email = None
```

wenn send_results = False müssen die Einstellungen nicht gemacht werden und die Bilder werden lokal gespeichert.

Autostart

Um die Anwendung beim booten des Raspberry's automatisch zu starten:

Im *launcher.sh* den pfad zum *main.py* Script definieren.

Anschließend service anlegen, der launcher.sh beim booten ausführt

```
sudo nano /lib/systemd/system/my_init.service
```

und folgendes schreiben:

```
[Unit]
Description=init
After=multi-user.target

[Service]
User=pi
Group=pi
Type=idle
ExecStart=bash /pfad/zu/launcher.sh &

[Install]
WantedBy=multi-user.target
```

Zugriffsrechte geben:

```
sudo chmod 644 /lib/systemd/system/my_init.service
```

Daemon neu laden (nach jeder änderung)

```
sudo systemctl daemon-reload
```

manuelles starten/stoppen des Service

```
sudo systemctl start my_init.service  
sudo systemctl stop my_init.service
```

automatisches starten beim Boot aktivieren/deaktivieren

```
sudo systemctl enable mysript  
sudo systemctl disable mysript
```

Status abfragen (gibt Konsolenausgabe von main script aus)

```
sudo systemctl status my_init.service
```

OpenVino Installationen

Ubuntu

1. [OpenVino](#) herunterladen und entpacken

```
cd ~/Downloads  
tar -xvzf l_openvino_toolkit_p_<version>.tgz
```

2. OpenVino und dependencies installieren

```
cd l_openvino_toolkit_p_<version>  
sudo ./install_GUI.sh  
cd /opt/intel/opencvino/install_dependencies  
sudo -E ./install_openvino_dependencies.sh
```

3. Environment Variablen setze

```
source /opt/intel/opencvino/bin/setupvars.sh
```

(in .bashrc hinzufügen für permanent)

4. ModelOptimizer installieren

```
cd
/opt/intel/opencv/deployment_tools/model_optimizer/install_prerequisites
sudo ./install_prerequisites.sh
```

5. NCS2 einrichten

```
sudo usermod -a -G users "$(whoami)"
sudo cp /opt/intel/opencv/inference_engine/external/97-myriad-usbboot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig
sudo reboot
```

Raspberry Pi

1. Raspberry Pi Setup

[Rasbian](#) herunterladen und img sd karte mit zb [BalenaEtcher](#) erstellen.

Dann ssh datei auf sd karte anlegen:

```
cd media/<user>/boot
touch ssh
```

SD Karte in Raspberry und Verbindung über Ethernet Kabel herstellen.

Dann:

```
ssh pi@raspberrypi.local
```

passwort: *raspberry* eingeben.

2. OpenVino auf Raspberry Pi

[OpenVino](#) und [OpenCV](#) installieren.

Mobiles Internet

1. prüfen ob stick erkannt wird:

```
lsusb
```

ausgabe:

```
Bus 001 Device 004: ID 12d1:14dc Huawei Technologies Co., Ltd. E33372  
LTE/UMTS/GSM HiLink Modem/Networkcard
```

2. Browser öffnen und auf 129.168.8.1 gehen
3. Pin eingeben (Pin prüfung deaktivieren)

als standart Verbindung fenstlegen, da sonst verbindungswechsel stattfinden kann und dann connection script evtl nicht mehr läuft.

```
route -n # aktuelle metrik rausfinden
```

```
ifmetric eth1 100 # niedriger als die anderen für höhere proi
```

für permant:

in

```
sudo nano /etc/dhcpd.conf
```

hinzufügen:

```
interface eth1  
metric 200 # niedrigste  
static ip_address=192.168.8.1
```