MEC308

Industrielle Robotik I

Benutzerhandbuch von Echtzeitig Objekt und Farberkennung der Tasse und der Flasche durch maschinelles Lernen Projekt

TMS

GRUPPE 5

GRUPPENMITGLIEDER :

Mustafa Barış Özer	150501104
Can DOĞAN	150501129
Koray IŞIK	150501121
Yunus Emre TELLİOĞLU	150501115

MODULVERANTWORTLICHER :

Dr.-Ing. Soner EMEC

Türkisch-Deutsche Universität, Istanbul

Fakultät für Ingenieurwissenschaften

Anweisungen zum Ausführen des Codes

Installation von Miniconda, Python, OpenCV, PyCharm

Python v3

Wir werden conda (zur Installation von Python v3) verwenden, ein Open Source-Paket- und Umgebungsverwaltungssystem, das unter Linux, Windows und macOS ausgeführt wird. Ein kostenloses Minimal-Installationsprogramm für conda ist Miniconda, das nur conda selbst, Python und eine kleine Anzahl anderer nützlicher Pakete enthält.

Nach der Installation von Miniconda können Sie mit conda alle anderen Pakete installieren und Umgebungen erstellen. Beispiel zum Erstellen einer Umgebung mit Python v3.6:

conda create -n name-of-the-environment python=3.6

Beispiel zum Auflisten aller vorhandenen Umgebungen:

conda env list

Beispiel zum Aktivieren der Umgebung:

conda activate name-of-the-environment

Anweisungen zur Installation von Miniconda auf Windows:

1. Laden Sie hier (https://docs.conda.io/en/latest/miniconda.html) das Installationsprogramm von Miniconda von der offiziellen Website herunter (wählen Sie das für Windows und Python v3).
2. Öffnen Sie den Explorer und suchen Sie die heruntergeladene Installationsdatei
3. Führen Sie die Installationsdatei aus, indem Sie darauf doppelklicken. Die Datei sollte so oder ähnlich aussehen:

Miniconda3-latest-Windows-x86_64.exe

4. Befolgen Sie die Anweisungen auf den Installationsbildschirmen
5. Öffnen Sie nach der Installation die Anaconda Prompt über das Startmenü
6. Überprüfen Sie die Installation, indem Sie den folgenden Befehl in Anaconda Prompt ausführen:

conda list

OpenCV

Es gibt nur wenige der einfachsten Möglichkeiten, OpenCV zu installieren:

• mit conda

• mit pip für Python v3

Bei der ersten Option (conda) muss conda installiert sein und einen der folgenden Befehle verwenden:

- `conda install -c conda-forge opencv`
- `conda install -c conda-forge/label/gcc7 opencv`
- `conda install -c conda-forge/label/broken opencv`
- `conda install -c conda-forge/label/cf201901 opencv`

Verwenden Sie mit der zweiten Option (pip) einen der folgenden Befehle:

- `pip install opencv-python`
- `pip install opencv-contrib-python`

Oder

- `pip3 install opencv-python`
- `pip3 install opencv-contrib-python`

Passt auf! Nicht alle Versionen von Python sind mit verschiedenen OpenCV-Distributiven kompatibel.

Die für diesen Kurs installierte Kombination lautet wie folgt:

- `conda create -n name-of-the-environment python=3.6`
- `conda activate name-of-the-environment`
- `pip install opencv-python`

OpenCV wird in der ausgewählten Umgebung installiert. Um zu überprüfen, ob OpenCV installiert wurde,

Führen Sie Python in einer beliebigen Form aus (z. B. aktivieren Sie einfach Ihre mit conda erstellte Umgebung und geben Sie python ein.) und verwenden Sie die folgenden zwei Codezeilen:

- `import cv2`
- `print(cv2.__version__)`

Infolgedessen muss die Version von installiertem OpenCV wie folgt angezeigt werden: 4.1.0

Sie können die Installation auch überprüfen, indem Sie nur eine Codezeile in Terminal (oder Anaconda Prompt) ausführen. Vergessen Sie jedoch nicht, Ihre Umgebung zu aktivieren:

```
python -c "import cv2; print(cv2.__version__)"
```

Anweisungen zur Installation von Pycharm auf Windows:

Anweisungen zur Installation der kostenlosen Open-Source-IDE für die reine Python-Entwicklung und der Community-Version von Pycharm.

Anweisungen zur Installation von PyCharm unter Windows:

1. Laden Sie hier das Installationsprogramm von PyCharm von der offiziellen Website herunter (klicken Sie auf die Schaltfläche für die Community-Version).

2. Öffnen Sie den Explorer und suchen Sie die heruntergeladene Installationsdatei

3. Führen Sie die Installationsdatei durch Doppelklick aus. Die Datei sollte so oder ähnlich aussehen:

pycharm-community-2019.2.4.exe

4. Befolgen Sie die Anweisungen auf den Installationsbildschirmen

Anweisungen zur Coco Dataset

Sie können im Projekt Ordner tranierte Coco Dataset finden. Coco Dataset soll im gleichen Ordner mit Projekt Ordner sein.

Erklärung des Codes

Importierung von Bibliotheken

Importieren Sie die folgenden Python 3-Bibliotheken in die Datei yolo-3-camera.py

```
import numpy as np
import cv2
import time
camera = cv2.VideoCapture(0)
```

Um den Code auszuführen, bewahren Sie die Dateien coco.names, yolov3.cfg, yolov3.weights unter dem Ordner yolo-coco-data im selben Ordner wie yolo-3-camera.py auf.

Erklärung des Codes Schritt für Schritt

1. Platzieren Sie den Zugriffspfad der Datei, in der die Klassen im Code registriert sind.

```
with open('yolo-coco-data/coco.names') as f:
    labels = [line.strip() for line in f]
```

2. Geben Sie Zugriffspfade für das fertige COCO-Dataset oder das von Ihnen trainierte Dataset .cfg- und .weights-Dateien an.

```
network = cv2.dnn.readNetFromDarknet('yolo-coco-data/yolov3.cfg',
                                     'yolo-coco-data/yolov3.weights')
```

3. Bestimmen Sie die minimale Konfidenzwert und den Schwellenwert.

```
probability_minimum = 0.5
threshold = 0.3
```

4. Synchronisieren Sie die Variable "Frame" mit den Bildwerten, die von der Kamera stammen.

```
    _, frame = camera.read()
```

5. Passen Sie die Fenstergröße an.

```
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
                                 swapRB=True, crop=False)
```

6. Erstellen Sie einige Array , die in Zukunft erstellt werden sollen.
    - “bounding_boxes” Array enthält die Boundig Boxes
    - “confidences” Array enthält die Konfidenzwerte.
    - “class_numbers” Array enthält die Klassennummern der erkannten Objekte.

```
    bounding_boxes = []
    confidences = []
    class_numbers = []
```

7. Es zeigt, welche Klasse zur höchsten Konfidenzrate des erkannten Objekts gehört.

```
        for detected_objects in result:

            scores = detected_objects[5:]
            class_current = np.argmax(scores)
            confidence_current = scores[class_current]
```

8. Wegen der Benutzung von COCO Dataset, gibt es 80 verschiedene Klassen. In dieser Projekt brauchen wir nur 2 (Tasse und Flasche).Deswegen hier 41 und 49 als class_current benutzt werden. confidence_current > probability_minimum zeigt, dass Konfidenzrate soll größer als 50%.

```
if confidence_current > probability_minimum and class_current == 41 or class_current == 39  :
```

9. "Non-Maximum Suppression Technik" wird angewendet. Dank dieser Technik werden Bounding Boxen mit geringer Wahrscheinlichkeit, die derselben Region entsprechen, entfernt.

```
results = cv2.dnn.NMSBoxes(bounding_boxes, confidences,
                           probability_minimum, threshold)
```

10. Platzieren Sie die Koordinatenwerte des erkannten Objekts in den Bounding Boxen.

```
x_min, y_min = bounding_boxes[i][0], bounding_boxes[i][1]
box_width, box_height = bounding_boxes[i][2], bounding_boxes[i][3]
```

11. Aus dem Bild "Frame" wird nur der Teil des gerahmten "bounding_boxes" als separater Frame2 abgeschnitten. Die durchschnittlichen Farbwerte werde im ‘durchschnittliche _farbe_satir’ für jede Zeile im Bild frame2 behaltet. Die durchschnittliche Werte im 'durschnittliche_farbe_satir' Array wird genommen, dh der Durchschnitt des gesamten Bildes.

```
frame2=frame[y_min:box_height, x_min:box_width]
durschnittliche_farbe_satir = np.average(frame2,axis= None)
durschnittliche_farbe = np.average(durschnittliche_farbe_satir, axis= None)
```

12. Geben Sie die Farb-, Dicken- und Koordinatenwerte des Felds ein.

```
cv2.rectangle(frame, (x_min, y_min),
              (x_min + box_width, y_min + box_height),
              durschnittliche_farbe, 2)
```

13. Bestimmen Sie der Textgröße und des Schrifttyps.

```
cv2.putText(frame, text_box_current, (x_min, y_min - 5),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, durschnittliche_farbe, 2)
```

14. Hier können Sie die Exit-Taste einstellen. In diesem Projekt wurde es als “q” bestimmt.

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