Anwenderdokumentation:

Objekterkennung

Deep Learning Traffic Sign

Zeitraum: WS 2021/2022

Abgabedatum 10.01.2022

Projektteam
Nico Isheim
Fabian Beckdorf
Jacob Prütz
Ali Reza Teimoury
Julian Müller
Michael Sievers

Inhaltsverzeichnis

Voraussetzung

Für die Verwendung der Objekterkennung oder weiteren programmierung, wird das Programm Matlab benötigt. Dieses benötigt eine kostenpflichtige Lizenz, bzw. eine Hochschullizenz.
Zudem wurden bei uns Add-Ons verwendet. Diese werden benötigt um weitere Funktionalitäten in Matlab bereitzustellen. Folgende Add-Ons müssen daher installiert werden:

- Deep Learning Toolbox
- Deep Learning Toolbox Model for ResNet-50 Network
- Deep Learning Toolbox Model for AlexNet Network
- Parallel Computing Toolbox
- Image Processing Toolbox

Um die Netze mit den voreingestellten Einstellungen trainieren zu können, wird bei Verwendung von Hardwarebeschleunigung eine dedizierte Grafikkarte mit mindestens 8Gb Grafikspeicher benötigt.

Übersicht der Skripte

Skript	Kurzbeschreibung
LoadandRandomizeData	Lädt die aufgenommenen Bilder und deren ‘GroundTruth’. Die Bilder werden dann Zufällig zu 60% in Trainingsdaten, 10% in Validierungsdaten und 30% in Testdaten aufgeteilt.
augmentData	Hier wird eine Datenerweiterung angewendet. Die Datenmenge wird durch hinzufügen modifizierter Kopien bereits vorhandener Daten erhöht (z.b. durch spiegeln, drehen, zoomen etc.).
preprocessData	Formatiert die Bilder aus dem übergebenen DataStore in die übergebene Bildgröße.
Step1TrainRegionDetection Step2TrainRegionDetection	In diesem Script wird ein resnet50 CNN Netz verwendet um ein Faster R-CNN zu erstellen, dass die Verkehrsschilder aus den Bildern erkennt.
Step1TestRegionDetection Step2TestRegionDetection	In diesem Script wird das trainierte resnet50 Faster R-CNN Netz mit 30% der Bilder getestet.
Step1TrainClassification Step2TrainClassification	In diesem Script werden die erkannten Schilder eingelesen und in Training-, Validierung- und Testdaten aufgeteilt. Danach wird ein CNN antrainiert, welches die Schilder klassifizieren soll.
Step1TestClassification Step2TestClassification	In diesem Script wird das trainierte CNN Netz mit 30% der GBS-Bilder getestet.

Verwendung

Falls noch bestimmte Ressourcen fehlen, wie trainierte Netze oder ausgeschnittene Schilder, müssen die Skripte in folgender Reihenfolge aufgerufen werden.

1. Step2TrainRegionDetection
2. Step2TestRegionDetection
3. Step2TrainClassification
4. Step2TestClassification

Diese Reihenfolge gewährleistet, dass alle benötigten Ressourcen zur Ausführung der jeweiligen Scripte zur Verfügung stehen. Nach Vollständiger Ausführung, können die Skripte auch einzeln aufgerufen werden. Zudem gibt es noch das Skript Step2TestSystem, wo diese Skripte in genau dieser Reihenfolge aufgerufen werden. Dies erleichtert es einen Aufruf zu starten, um das ganze System zu testen.

Dateien

Die Dateien sind Aufgeteilt in

- Pictures — Bilder der Geschwindigkeitsbegrenzungsschilder
- Labels — Bilder der ClassLabels der GBS
- Neuronale Netze — Trainierte Netze
- Step1 — Hauptskripte
- Step2 — Optimierte Hauptskripte
- Funktionen — Ausgelagerte Funktionen die in den Hauptskripten verwendet werden
- SignsFound — Hier werden die ausgeschnittenen Bilder der Schilder gespeichert

Dementsprechend ist die Ordnerstruktur aufgebaut.