

STATUS-UPDATE 03

BISHERIGE FORTSCHRITTE:

Seit dem letzten Status-Update

1. Meeting zur Vorstellung von CVAT und dem ersten Prototyp
2. Einrichtung GitHub-Projekt mit Source Code und Anleitungen
3. Konzept und Skript zur Strukturierung und Filterung der großen Menge an Bilddaten erstellt
4. Konzept zur Benennung der zu unterscheidenden Klassen erarbeitet
5. Entwurf der Annotationsrichtlinie erstellt
6. Neuer Beispieldatensatz erstellt durch manuelle Annotation mehrerer Klasse

NÄCHSTE SCHRITTE:

1. Untersuchen des unvollständigen CVAT Exports
2. Test des Prototyps mit dem neuen Beispieldatensatz
3. Meeting zur Installation von CVAT und dem Prototyp bei Ihnen
4. Evaluierung verschiedener vortrainierter Modelle
5. Logging & Visualisierung des Trainingsverlaufs

BENÖTIGTES FEEDBACK:

1. Liste aller zu unterscheidenden Verkehrszeichen (gemäß erarbeitetem Konzept)
2. (Einrichtung CVAT & Prototyp bei Ihnen)
3. Annotationsrichtlinie verfeinern / absegnen
4. Schriftliche Freigabe Veröffentlichung des Beispieldatensatzes / der Bilder im Repo

STATUS-UPDATE 03

ANHANG

1. Termin 08.11.2024: Vorstellung CVAT & Prototyp

Das Computer Vision Annotation Tool (CVAT) sowie der erste Prototyp wurde präsentiert.

Einige der besprochenen Punkte:

- Der mir bereitgestellte Datensatz entspricht ca. einem hundertstel der Gesamtmenge
- Die Daten sind in Ordnern strukturiert, wobei die obere Ebene den Straßenabschnitt bezeichnet und die darunterliegenden Ordner den Kreuzungsbereich
- Aufeinanderfolgend-nummerierte Dateien mit dem gleichen Dateinamen-Anfang (z.B. SGA000001) sind nacheinander aufgenommene Bilder einer Fahrt und vermutlich ähnlich zueinander. Um die Datenmenge zu reduzieren, sollte es reichen, hier nur jeweils ein Bild je Serie zu verwenden.
- Datensatz enthält etwa 660.000 Bilder. Für das Training sollten mindestens 10.000 Bilder verwendet werden, wobei die genaue Anzahl abhängt von der Anzahl zu unterscheidender Typen von Verkehrszeichen.
- Der Datensatz enthält auch seltene Schilder, beispielsweise im Hafengebiet. Auch diese sollten idealerweise erkannt werden.
- In Anschluss an das Meeting wird eine Installationsanleitung für CVAT und den Prototyp sowie entsprechender Quelltext bereitgestellt.
- Es ist okay, beispielhaft Bilder des Datensatzes in unserem Repository zu veröffentlichen, sofern auf diesen keine Personen oder KFZ-Kennzeichen zu sehen sind.

2. Einrichtung GitHub Projekt

Um dem entwickelten Quelltext und die Installationsanleitungen für CVAT und den Prototyp bereitzustellen wurde ein GitHub Repository unter der folgenden Adresse erstellt:

<https://github.com/AlexanderMelde/TrafficSignRecognition>

Zum aktuellen Stand ist dieses Repository nur auf Einladung abrufbar.

Das GitHub Konto von Tim Balschmiter wurde bereits eingeladen, weitere Konten können auf Anfrage ebenfalls hinzugefügt werden.

In dem Repository finden sich zum aktuellen Zeitpunkt Quelltexte, Installationsskripte, Dokumentation und Anleitungen, unterteilt anhand der verschiedenen Schritte und Teilbereiche:

- *annotation* (Hilfsskripte und Beispieldateien für das Beschriften mittels CVAT)
- *datasets* (Ein Ordner mit Beispiel-Datensatz, wie er von CVAT exportiert werden sollte)
- *docs* (Unterstützende Dateien (Bilder) für die Readme-Datei)
- *preprocessing* (Skript zur Vorverarbeitung der Bilddaten, siehe unten)
- *recognition* (Programm für das Training und die Inferenz von KI-Modellen)
- *README.md* (Anleitungen zur Bedienung des Repos und der Installation)

STATUS-UPDATE 03

3. Skript zur Vorverarbeitung der Bilder

Um die großen Datenmengen auf das wichtigste zu begrenzen und übersichtlich zu strukturieren wurde ein Skript entwickelt, dass die folgenden Funktionen erfüllt:

1. Filtern aufeinanderfolgender Bilder der gleichen Straße (zu ähnliche Bilder)
2. Durchmischen der Bilder
3. Aufteilen der Bildmenge in die drei Splits „train“, „val“ und „test“
4. Kopieren der Bilder in eine neue, flachere Ordnerstruktur

Das Skript arbeitet mit der Ordnerstruktur der Beispieldaten. Diese Bilder sind in Ordnern der Art:

```
Photos/0770079603/050_052_/SXA000053.jpg
```

Diese Bilder werden gefiltert und kopiert an einen neuen Ordner-Pfad der Art:

```
Photos_filteres/train/0770079603_050_052_SXA000053.jpg
```

Mit dieser Struktur ist ein einfaches importieren der Bilder in CVAT möglich, wobei dennoch die Informationen über die Originalpfade der Bilder nicht verloren gehen.

Die Ordnerpfade sowie das Verhältnis der verschiedenen Splits ist konfigurierbar.

Als Standardwert wurden 70% für Training, 20% für Test sowie 10% für Validierung vorgesehen.

Das Skript befindet sich im Repository unter diesem Pfad:

```
preprocessing/restructure_source_images.py
```

STATUS-UPDATE 03

4. Konzept zur Benennung der Klassen von Verkehrszeichen

Um die voraussichtlich hohe Anzahl zu unterscheidender Typen von Verkehrszeichen zu strukturieren ist ein Konzept notwendig.

Ich schlage vor, aus der folgenden Liste:

[„Bildtafel der Verkehrszeichen in der Bundesrepublik Deutschland seit 2017“ \(Wikipedia\)](#)

bzw. der zugrundeliegenden [Anlagen 1–4 zu §40–43 STVO](#)

alle relevanten Schilder herauszusuchen und anhand des folgenden Schemas zu benennen:

[Nummer]_[vereinfachte Bezeichnung]

Beispielsweise:

- 205_vorfahrt_gewaehren
- 206_halt_vorfahrt_gewaehren
- 283_10_absolutes_halteverbot_anfang_rechts
- 314_10_parken_links
- parkplatz
- 265_hoehe

Verkehrszeichen-Nummern werden dabei mit Unterstrichen geschrieben und die Namen ebenfalls in „snake_case“ sowie mit ausgeschriebenen Umlauten sowie ohne Zahl am Anfang formuliert.

So ist sichergestellt, dass die Klassennamen sowohl für Maschinen als auch Menschen lesbar ist.

Sollte das Verkehrszeichen keine Nummer haben, so wird diese weggelassen.

Das Erstellen dieser Liste erfordert Fachwissen zur STVO bzw. dem später gewünschten Einsatzszenario. Ich bitte daher Sie um die Erstellung dieser Liste. Bei der Erstellung dieser Liste überlasse ich es auch Ihnen, ob sie beispielsweise 274_hochstgeschwindigkeit anhand der Geschwindigkeits-Werte weiter unterteilen möchten in z.B. 274_hochstgeschwindigkeit_30, 274_hochstgeschwindigkeit_50 etc.

Dabei ist zu beachten, dass für das Training für jede dieser Klassen ausreichend viele Beispielbilder benötigt werden. Die genaue Anzahl ist abhängig von der Qualität des Pretrained-Modells, grob geschätzt wären aber rund 50+ (verschiedene) Beispiele je Schild gut, es gilt aber: Je mehr Bilder pro Klasse, desto besser.

Anhand einer einfachen Zeilengetrennten Liste dieser Klassen kann anschließend automatisiert eine „labels.json“ Datei erzeugt werden, die für CVAT benötigt wird.

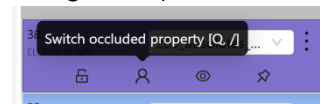
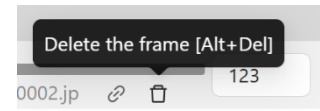
Hinweis: CVAT unterstützt für jede Klasse außerdem Attribute wie Nummern oder Freitexte, was ggf. sinnvoll sein könnte bei Geschwindigkeitsbegrenzungen oder Ortsschildern etc. Die Qualität der Daten-Annotation würde damit aufgewertet, aufgrund des erhöhten Aufwands bei der Annotation habe ich diese Variante aber erstmal nicht weiter berücksichtigt.

STATUS-UPDATE 03

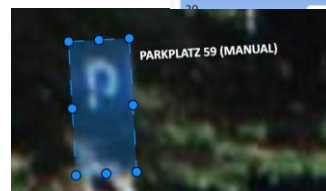
5. Annotationsrichtlinie (Entwurf)

Definiert die gewünschte Genauigkeit und den Umgang mit bestimmten Situationen.

1. Tasks in CVAT sollten in verschiedene Jobs unterteilt werden, z.B. mit je 100 Frames
2. Jobs in CVAT sollten alle drei Stages durchlaufen:
 - a. *annotation* – Person A
 - i. Bilder mit Gesichtern & Kennzeichen löschen
 - ii. Verkehrszeichen mit Rahmen markieren
 - b. *validation* – Person B
 - i. Prüfen ob Rahmen korrekt sind, wenn nicht: „Issues“ markieren
 - ii. Bilder mit Gesichtern & Kennzeichen löschen
 - c. *acceptance* – Person A
 - i. Issues durchklicken und lösen (z.B. fehlende Schilder ergänzen)
3. Wenn ein Verkehrszeichen zu mehr als ca. 10% verdeckt wird, sollte die *occluded* Eigenschaft gesetzt werden.

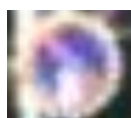


not occluded, „normale“ Abnutzung

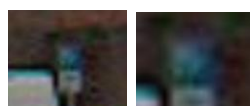


occluded, halbes Schild verdeckt durch Baum

4. Bei teilweise verdeckten Schildern wird die anzunehmende komplette Form gezeichnet, nicht nur der sichtbare Teil.
5. Schilder sollten markiert werden, wenn ein Mensch sie eindeutig erkennen kann, ohne raten zu müssen. Zu kleine oder anderweitig unscharfe Schilder werden nicht annotiert.



nicht annotiert, nicht eindeutig erkennbar ob unten ein Pfeil ist



nicht annotiert, Kontext vermutlich Parkplatz, es ist aber nichts erkennbar



annotiert, stark überbelichtet aber eindeutig erkennbar



annotiert, nur halb sichtbar aber eindeutig erkennbar

6. Zusammenhängende Schilder (z.B. Parkplatz mit weißem Zusatzschild) werden getrennt voneinander annotiert und bekommen somit mehrere Rahmen
7. Schilder, die nicht exakt zu einem der Label passen, werden nicht annotiert (kein „nächstbestes“ passendes Schild suchen).
8. Rahmen werden mit den jeweils passenden Annotationstools gezeichnet. Rundes Schild = Ellipse, dreieckig = Polygon (3 Corners), viereckig = Rechteck, bei Bedarf Formen drehen.
9. Nur stark schrägläufig verzerrte (skewed) Schilder benötigen ein Polygon statt Rechteck
10. Rahmen werden so nah wie möglich gezeichnet, ohne dass Teile des Schilds abgeschnitten werden. Lieber etwas zu viel im Rahmen als zu wenig.
11. Bilder ohne Verkehrszeichen werden nicht gelöscht (außer falls Datenschutz).
12. Rückseiten von Verkehrszeichen werden nicht markiert
13. Verkehrszeichen in Plakaten, auf Fahrzeugen oder in Reflektionen werden nicht markiert (für unseren Anwendungsfall unerwünscht).

STATUS-UPDATE 03

6. Neuer Beispieldatensatz: Beispielhafte Annotation anhand des neuen Konzepts

Die mir zur Verfügung stehenden Bilder wurden vorverarbeitet und mit den neuen Konzepten, der Annotationsrichtlinie und der Unterscheidung verschiedener Typen von Verkehrszeichen manuell annotiert hinsichtlich der folgenden Klassen beschriftet:

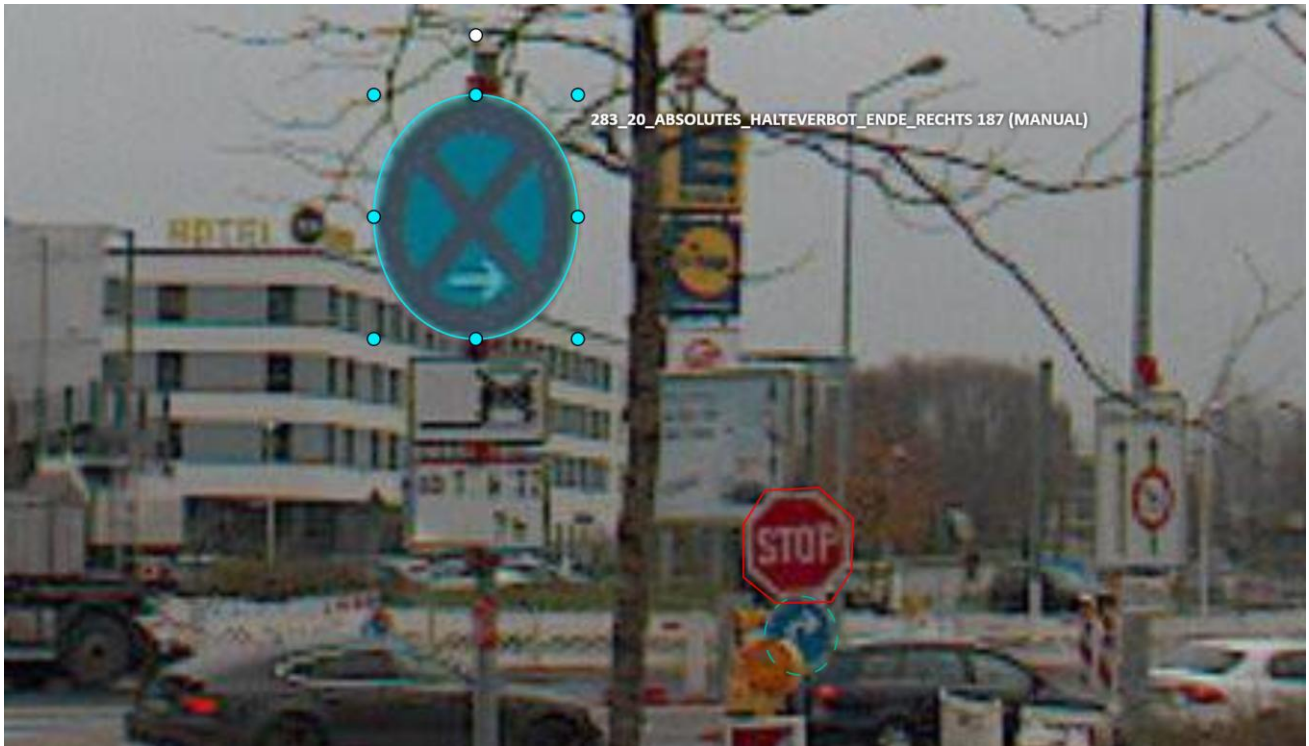


Mithilfe des neuen stats.py Skripts wurde ermittelt, welche Klassen wie oft annotiert wurden:

Zeichen	train 501 Bilder 218 Annot.	val 71 Bilder 22 Annot.	test 144 Bilder 73 Annot.
123_arbeitsstelle	3	0	0
205_vorfahrt_gewaehren	19	3	6
206_halt_vorfahrt_gewaehren	3	0	0
209_fahrtrichtung_rechts	3	0	1
209_30_fahrtrichtung_geradeaus	2	0	3
250_verbot_alle	1	0	2
265_hoehe	3	0	1
274_1_zone_30	0	0	0
274_hoehstgeschwindigkeit	0	0	0
283_10_absolutes_halteverbot_anfang_rechts	8	0	2
283_20_absolutes_halteverbot_ende_rechts	13	1	2
283_30_absolutes_halteverbot_mitte_rechts	3	0	1
283_absolutes_halteverbot	51	6	11
286_10_eingeschraenktes_halteverbot_anfang_rechts	0	0	0
286_20_eingeschraenktes_halteverbot_ende_rechts	2	1	0
286_30_eingeschraenktes_halteverbot_mitte_rechts	1	0	0
286_eingeschraenktes_halteverbot	13	1	2
314_10_parken_links	0	0	0
314_20_parken_rechts	0	0	0
314_30_parken_mitte	0	0	0
314_50_parkhaus	0	0	0
325_1_verkehrsberuhigter_bereich_beginn	1	0	0
325_2_verkehrsberuhigter_bereich_ende	0	0	0
parkplatz	0	0	0

STATUS-UPDATE 03

Es fällt auf, dass bei einigen Klassen (vor allem Parken) 0 steht, obwohl diese definitiv annotiert wurden. Hier scheint CVAT nicht alles zu exportieren, auch die Summe der Objekte entspricht daher nicht der Erwartung. Dieser Fehler muss noch geprüft werden.



Als nächstes erfolgt ein Test des Prototypens auf diesen Daten.