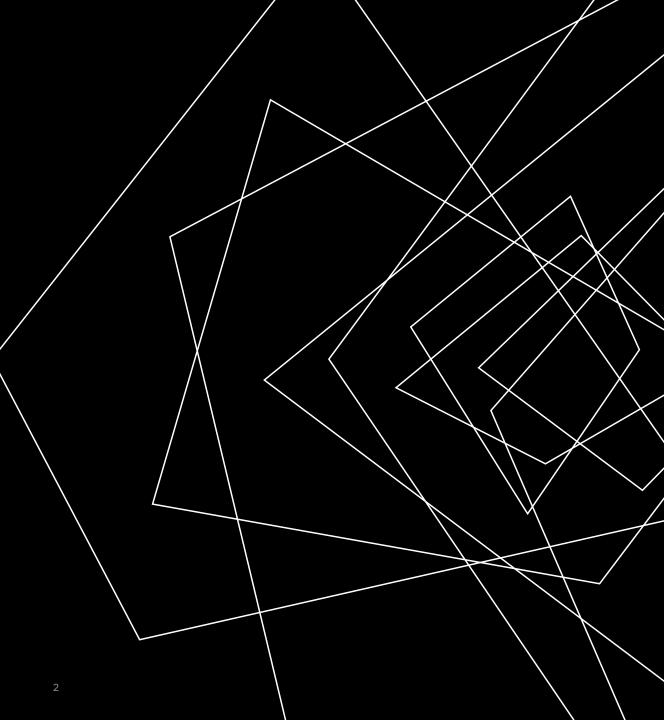


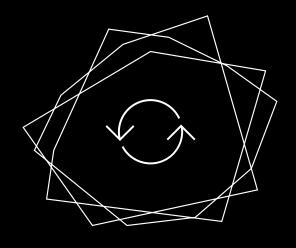
Alwin Zomotor, Ben Schlauch

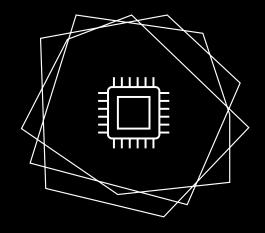
# AGENDA

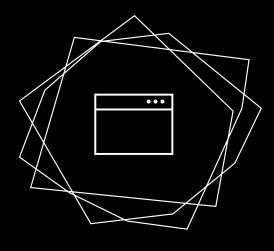
- 1. Mögliche Lösungen
- 2. Herausforderungen
- 3. Unser konventioneller Ansatz
- 4. Machine Learning
- 5. Kennzeichenerkennung



# LANE DETECTION - MÖGLICHE LÖSUNGEN







#### HOUGH TRANSFORM

Schlechte Performance Liefert wenig Punkte, nur Polynome 1. Grades konnten gut gefitted werden

**CANNY EDGE** 

Schneller als mit Hough Transform-Pipeline und Liefert bessere Ergebnisse

#### **SLIDING WINDOWS**

Theoretisch eine weitere Option

STÖRPUNKTE	Viele Störpunkte nach Vorverarbeitung erschweren das Polynomfitting
SPUREN TRENNEN	Linke und Rechte Spur trennen -> Besonders bei starken Kurven
POLYNOMFITTING	——————————————————————————————————————
KURVENRADIUS BERECHNEN	Keine Information, ob berechnete Werte Plausible sind.

### HERAUSFORDERUNGEN

#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

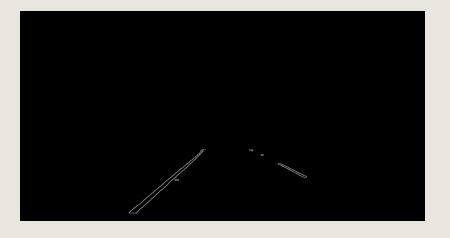
- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

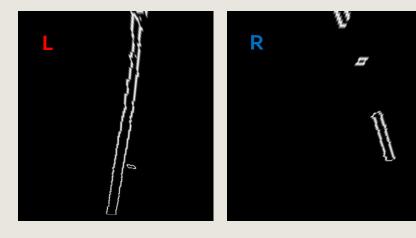
- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### Lane Detection Pipeline

- 1. Entzerren
- 2. Farben Filtern
- 3. In Grauwert-Bild umwandeln
- 4. Canny-Edge-Detection
- 5. Region of Interest
- 6. In Vogelperspektive Transformieren
- 7. Fahrspuren separieren
- 8. Kurven fitten
- 9. Radius berechnen

- Hough Transform Lines und Treshold
- Gelbe als linke und weiße als rechte Spur erkennen
- Horizontale Linien entfernen



#### **OPTIMIERUNG**

# PROGRAMM PROFILEN

 Eingebauter Python Profiler in Pycharm

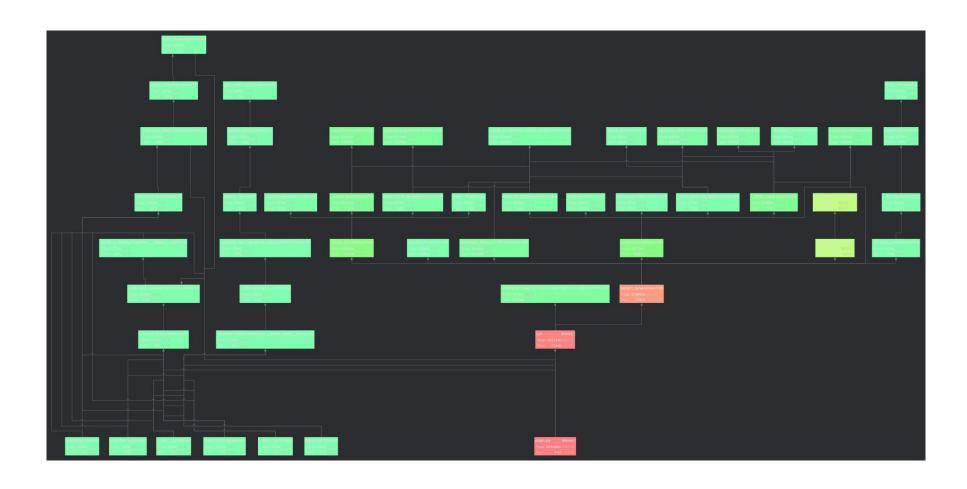
#### CALL GRAPH ANALYSIEREN

 Analysieren wo das größte Optimierungspotential liegt

# OPTIMIERUNG IMPLEMENTIEREN

• z.B. Parallelisierung

## **OPTIMIERUNG**



#### **OPTIMIERUNG**

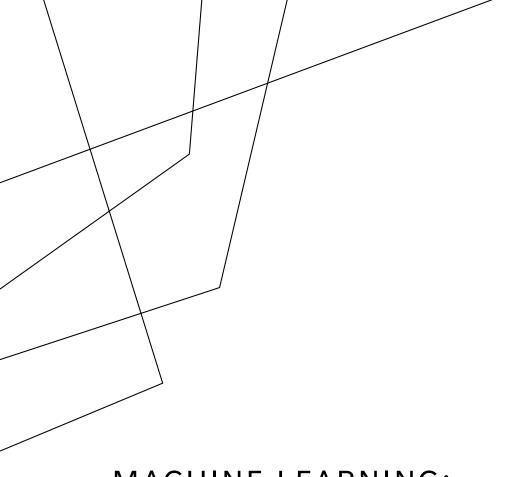
# <method'read'of'cv2.VideoCapture'object... &times;1261</p>

Total: **1602ms** 6,7 %

Own: 1602ms 6,7 %



→ Ohne Ausgabe ~54 FPS, mit Ausgabe ~32 FPS



MACHINE LEARNING: ULTRA FATS LANE DETECTION V2





# **ERKENNNUNG**

#### 1. OPENALPR

- Open Source
- End-to-End Lösung
- Eingabe: Bild
- Ausgabe: Text
- Sehr einfach
- Ergebnisse m\u00e4\u00dfig

#### 2. Eigene Pipeline

- 1. Kennzeichen finden
- 2. OCR -> Kennzeichen lesen





# **ERKENNNUNG**

# 1. KENNZEICHEN FINDEN MIT YOLO



2. OCR MIT KERAS\_OCR





# 3. TEXT VON LINKS NACH RECHTS ZUSAMMENSETZEN

Plate 0:

sam343

Plate 1:

sam323

# DANKE FÜR EURE AUFMERKSAMKEIT

Habt ihr Fragen?

