

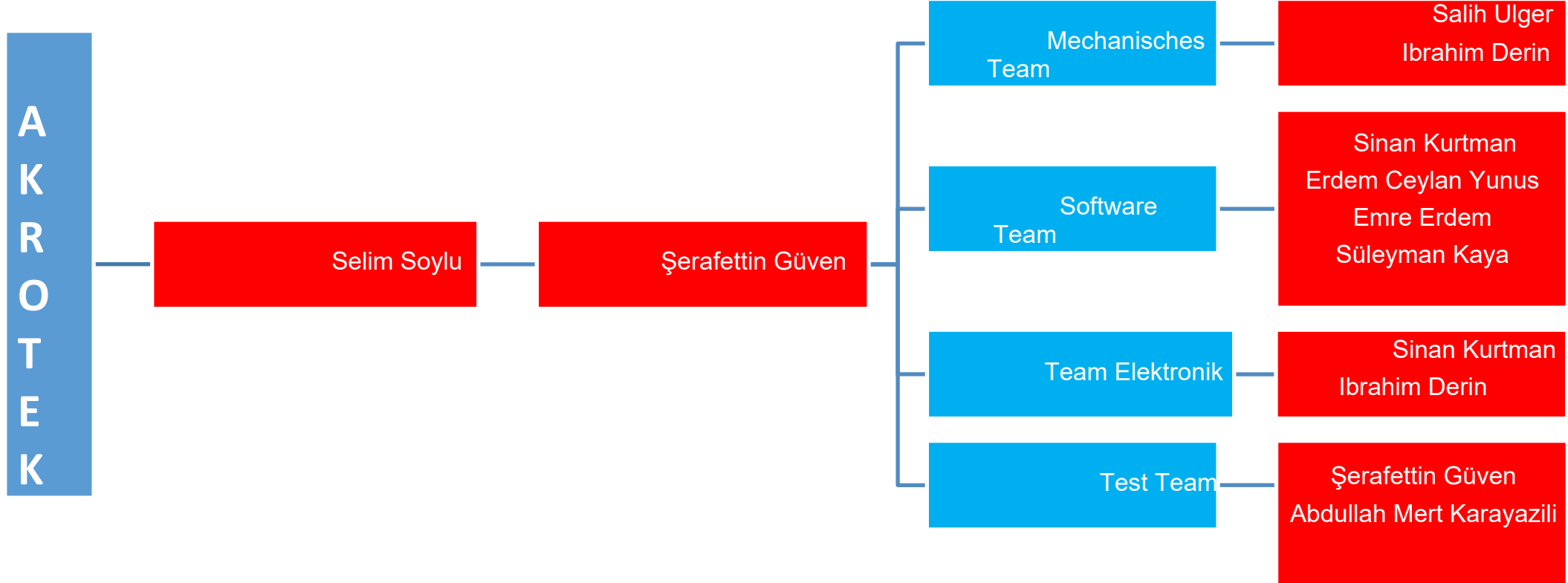
TEKNOFEST 2021

**Robotaxi Passenger Autonomous Vehicle
Competition Kategorie Originalfahrzeug Vorläufiger
Designbericht und Simulationspräsentation
#AKROTEK# #53614**

Team Organisation

Selim Soylu	Aksaray Universität	Akademischer Berater
Şerafettin GÜVEN	Aksaray Universität	EEM
Süleyman KAYA	Hazim Kulak Anatolische Gymnasium	Gymnasiast
Erdem CEYLAN	Universität Duzce	Computertechnik
Sinan KURTMAN	Aksaray Universität	EEM
Yunus Emre ERDEM	Urgup Tesan Wissenschaftliches Gymnasium	Gymnasiast
Abdullah Mert KARAYAZILI	Abdulhamid Khan Science High School	Gymnasiast
Salih ÜLGER	Abdulhamid Khan Science High School	Gymnasiast
Ibrahim DERIN	Aksaray Universität	EEM

Team Organisation





Zusammenfassung



Unser Fahrzeug mit autonomer Fahrfunktion wird die von den Kameras empfangenen Sofortbilder dank Computer Vision verarbeiten und die als Ergebnis dieser Verarbeitung erhaltenen Befehle an die Lenk- und Antriebsmotoren des Fahrzeugs senden, um die korrekte Ausrichtung des Fahrzeugs zu gewährleisten. Zu diesem Zweck wurden im Fahrzeug auch Hilfseinheiten wie ein LIDAR-Sensor, eine IMU-Einheit und ein Kurzstreckenradar verwendet. Daher wurde ein Algorithmus zur Sensorsplattung entwickelt, um eine Sensorkonsistenz zwischen diesen Sensoren herzustellen.

Die Karosserie des Fahrzeugs besteht aus Eisenprofilen und Decota wird als Beschichtungsmaterial verwendet, um Gewicht zu sparen. Die Struktur des Fahrzeugs wurde vollständig mit dem SolidWorks Modellierungsprogramm entworfen. Abgesehen von den Rädern und der elektrischen Servolenkung des Fahrzeugs wurden die Poryas, die Aufhängung und die Trägersysteme des Fahrzeugs von unserem Team entworfen.

Als wir mit diesem Projekt begannen, haben wir uns nach Kräften bemüht, einige Unzulänglichkeiten im Bereitschaftsgrad der Teammitglieder mit den Schulungen, die wir von verschiedenen Schulungsplattformen erhalten haben, und der Team-Solidarität zu beheben.

Damit unser Fahrzeug in der Lage ist, die in diesem Wettbewerb gestellten Aufgaben zu erfüllen, haben wir zunächst eine Simulation erstellt. Wir verfügen über das Wissen und die Ausrüstung, um die in der Simulation durchgeführten Operationen auf das echte Fahrzeug anzuwenden.

Im SCR hatte es geheißen, dass später entschieden werden würde, ob die Anzahl der Kameras des Fahrzeugs zwei oder drei sein würde. In Anbetracht unserer Diskussionen und der Aufgaben, die dem Fahrzeug zugewiesen wurden, wurde beschlossen, die Anzahl der Kameras auf vier zu erhöhen und dabei die Aufgabe des Einparkens zu berücksichtigen, die eine Rückfahrkamera erfordert. Natürlich wird die Bildverarbeitung von vier Kameras nicht gleichzeitig stattfinden. Welche Kamera verwendet wird, wird von der Steuerungssoftware ausgewählt und die Kameraauswahl erfolgt über einen Videowahlschalter (analoger Multiplexer).

Wie bereits erwähnt, wird das Fernsteuerungssystem des Fahrzeugs mit nRF24L01+PA Modulen realisiert.

Das Steuerhandgerät wurde von unserem Team mit Hilfe von SolidWorks-Zeichnungen entworfen und sein Gehäuse wurde durch Biegen von Plexiglas vorbereitet. Das Handgerät verfügt über einen Joystick zur Steuerung, eine Batterieanzeige und einen Not-Aus-Knopf.

Abgesehen davon wurden keine weiteren Änderungen an der Hardware vorgenommen.

Fahrzeug Spezifikationen



Das elektrische Lenkgetriebe des Fahrzeugs ist wie in der Abbildung dargestellt. Allerdings wird diesem Lenkgetriebe ein Encoder zur Positionserfassung hinzugefügt. Die vom Encoder empfangenen Informationen werden zur Berechnung des Fehlereingangs für den PID-Regler verwendet, der für die Richtungsstabilisierung des Fahrzeugs eingesetzt wird.

Das Trommelbremssystem, das mit dem BLDC-Motor verbunden ist, den wir zum mechanischen Bremsen v
Seil verbunden.

Da das Bremsen durch Ziehen erfolgt, wird das Seil mit einem 60 kg schweren Servomotor gezogen.



Der Informationsfluss zwischen dem NVIDIA Jetson Nano Minicomputer, auf dem die Bildverarbeitung stattfindet, und der STM32duino-Steuerkarte, die als zusätzliche Steuerkarte dient, über einen seriellen Kommunikationsanschluss mit 57600 Baud und im JSON-Datenformat.



Fahrzeugspezifikationen (Fortsetzung)

Das verwendete JSON-Format unten angegeben. Die IMU-Sensordaten werden von der Software auf der STM32duino-Steuerkarte verwendet, aber die Servo-Ausgangsdaten werden von der NVIDIA Jetson Nano-Karte an die Steuerkarte gesendet.

```
////////////////////////////////////  
////////// JSON FORMAT OUTPUT //////////  
////////////////////////////////////
```

```
// IMU-Sensorwerte
```

```
Printf("{\"Nicken\": euler.y(), \"Rollen\": euler.z(), \"Gieren\": euler.x()}"); Davon  
wird die Gierachse verwendet, um die Richtung des Fahrzeugs zu bestimmen.
```

```
// Servo-Ausgänge
```

```
Printf("{\"servo0": 0, "servo1": 1, "servo2": 2, "servo3": 3, "servo4": 4, "servo5": 5, "servo6": 6, "servo7": 7});  
Servoausgänge steuern Ausgaben wie Lenkung, Geschwindigkeitsregelung und Bremsen des Fahrzeugs.
```

Original Komponenten

Die Karosserie wurde vom AKROTEK-Mechanikerteam entworfen (Bild 1) und die Bauphase begann mit der Glasfaserbeschichtung auf Decota.

Die Scheinwerfer des Fahrzeugs wurden von unserem Team modelliert und mit einem 3D-Drucker gedruckt, ebenso wie die Akzente der Fahrzeugfront überarbeitet entsprechend dem Manövrierwinkel des Wettbewerbsgebiets in Kurven (Abbildung 2)



Bild 1



Bild 2

Original Komponenten

Die elektronischen Karten des Fahrzeugs wie BMS, PDB und Lenkungssteuerung (Visual 3) werden vom Elektronik-Team in einer virtuellen Umgebung entworfen, produziert und im Fahrzeug installiert. Neuronales Netzwerk über Tensorflow- und/oder pytorch-Bibliotheken für das Zeichenerkennungssystem des Fahrzeugs das Maschinenmodell wurde mit (Abbildung 4) trainiert

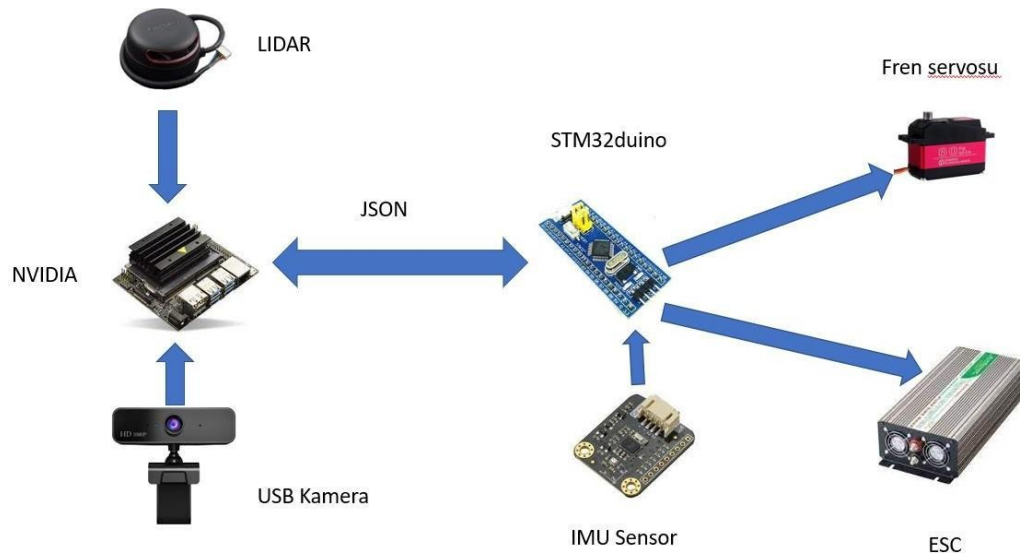
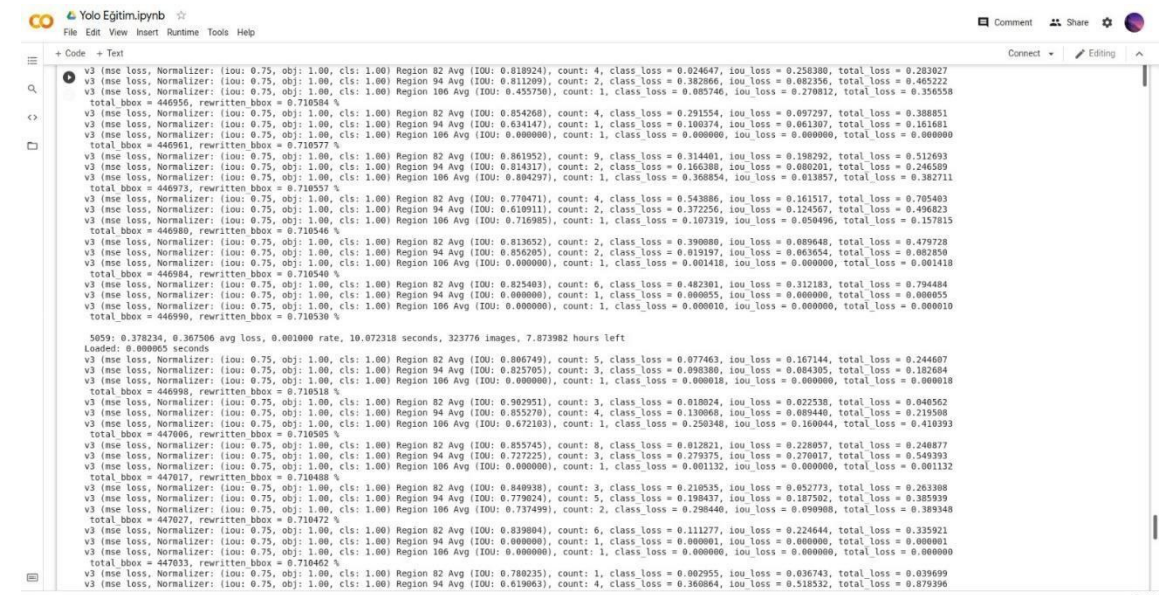


Bild 3
Bild 4



The screenshot shows the output of a YOLOv3 training session. The log displays various metrics for different regions (82, 94, 106) and classes (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Key metrics include mAP (mean Average Precision), mF1 (mean F1 score), mIoU (mean Intersection over Union), and mRec (mean Recall). The log also shows the total loss and the number of images processed. The training process is running on a GPU, as indicated by the 'GPU' label in the top right corner of the notebook interface.

Der 2D-LIDAR-Sensor wurde für die zweidimensionale Kartierung von Hindernissen rund um das Fahrzeug bevorzugt.



An der Vorderseite des Fahrzeugs wurde ein Radarsensor mit einer Reichweite von 20 m angebracht.

Verzögerungen, die durch LIDAR entstehen können, werden verhindert.

Das Fahrzeug ist dann in der Lage, schneller zum Stehen zu kommen.



Das DFRobot IMU-Sensorkit SEN0253 wird für die korrekte Ausrichtung des Fahrzeugs verwendet. Es wird nur der Magnetometerteil dieses Sensor-Kits verwendet. Dieses Sensorkit bevorzugt, weil es präziser messen kann.



Im Fahrzeug werden vier Kameras mit USB-Ausgang verwendet. Zwei dieser Kameras werden vorne und zwei hinten angebracht. Gleichzeitig werden die beiden vorderen oder die beiden hinteren Kameras aktiv sein.



Fahrzeug-Steuergerät

Die Arduino FreeRTOS Software wird auf der STM32duino Steuerkarte verwendet, die die Steuereinheit des Fahrzeugs darstellt.

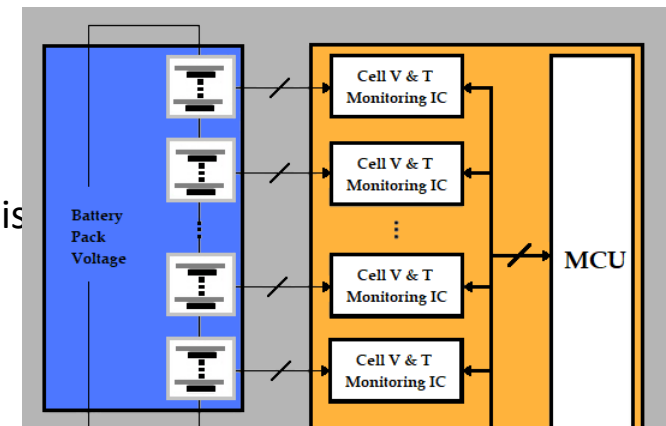
Auf diese Weise werden Multitasking und Real-Time Operating realisiert. Jede Aufgabe wird im Abstand von 1 ms ausgeführt

und Aufgaben können miteinander synchronisiert werden.



Die drahtlose Kommunikationseinheit nRF24L01+PA wird zwischen dem Fahrzeug und der Fernsteuerungseinheit eingesetzt. Zwei Arduino Nano-Controller werden zur Steuerung dieser Einheiten verwendet.

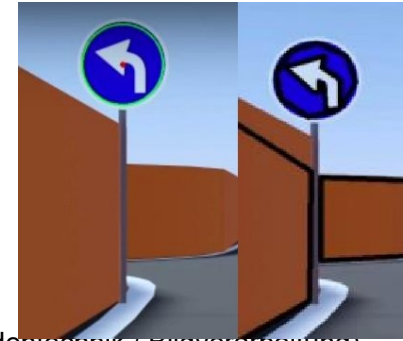
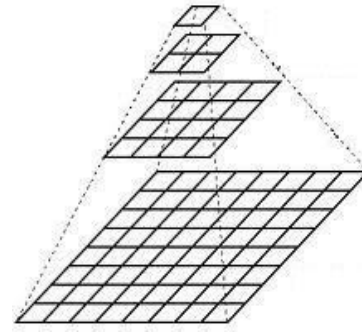
Um ein ausgeglichenes Laden und Entladen von Blei-Gel-Batterien zu gewährleisten



Das Batterie-Management-System wird verwendet.

Unser Algorithmus für autonomes Fahren kombiniert die folgenden Module mit Technologien für maschinelles Lernen und Bildverarbeitung mit der Ankunft eines Modells.

- Modul zur Erkennung von Fahrspuren und Barrieren
- Modul zur Erkennung von Schildern/Verkehrszeichen
- Modul zur Erstellung von Routen
- Fahr-Algorithmus
- Parken Algorithmus



(Gaußsche Pyramidenteknik / Bildverarbeitung)

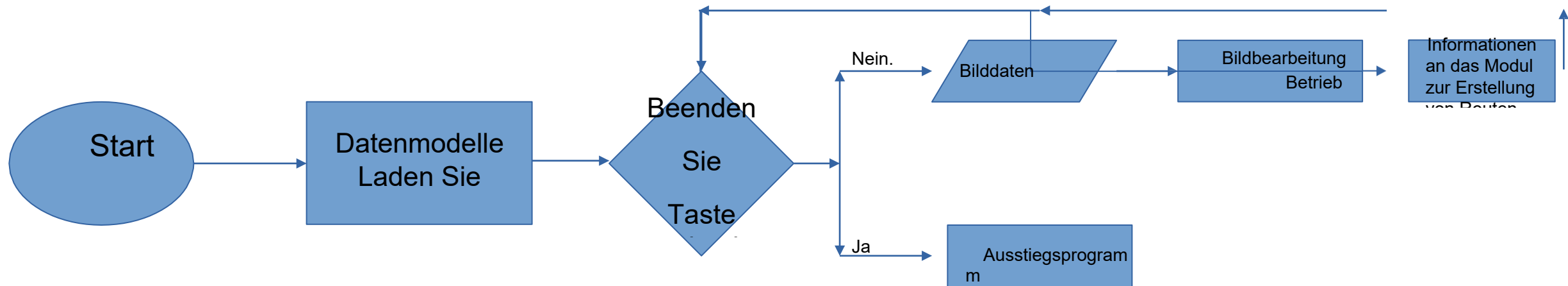
(Objekterkennung mit maschinellem Lernen)

Algorithmen für autonomes Fahren

Bei der Entwicklung von Modulen verwendete Technologien:

Bei der Entwicklung unserer Module wurden "Darknet/YOLO" aus den Technologien für maschinelles Lernen und "OpenCV-Contrib", "Numpy", "itemfreq" Bibliotheken für die Bildverarbeitung verwendet. Unser Modul und unser Algorithmus wurden in der Programmiersprache Python geschrieben.

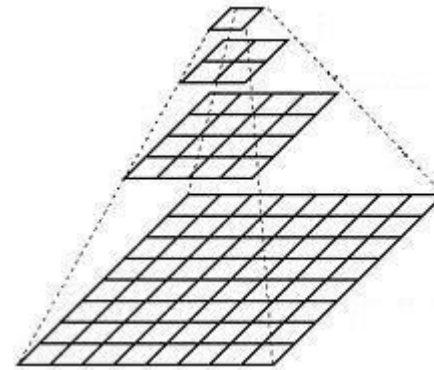
Der bei der Entwicklung der Module verwendete Algorithmus:



Techniken, die wir verwenden:

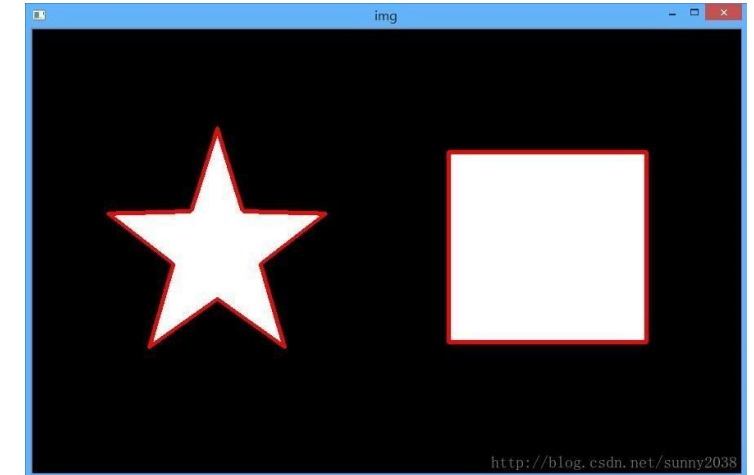
$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

(Bildverarbeitungsalgorithmus)



(Gaußsche Pyramide)

(OpenCV findContours Funktion)



Sicherheitsvorkehrungen

Mechanische Sicherheitsvorkehrungen

1-Physikalisches Notfall-Reaktionssystem



2-Ferngesteuertes Notrufsystem

3-Überhitzung

4-Kabel-Layout



Elektronische Sicherheitsmaßnahmen

1-Sicherheit der Stromversorgung

2-Schutz des Ladungseingangs



Simulation und Prüfung

Der Wettbewerbsstrack wurde mit dem Programm Blender erstellt.

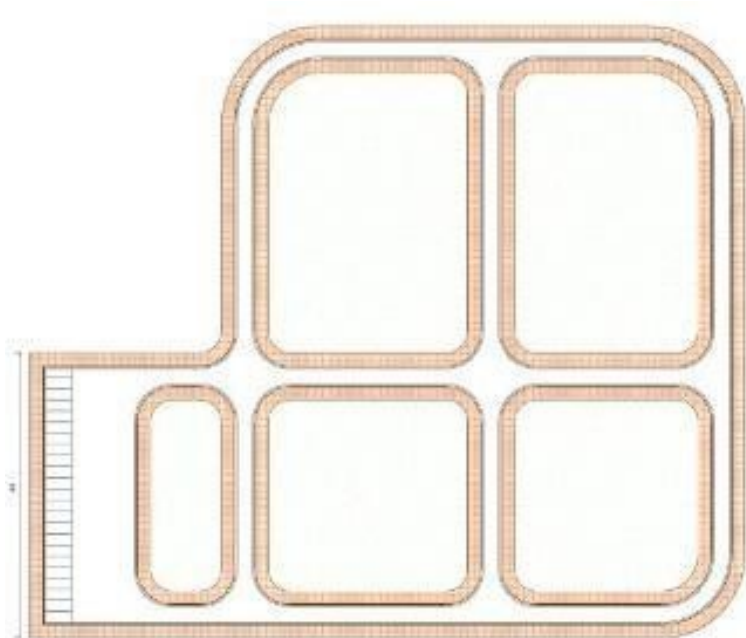


Bild 1

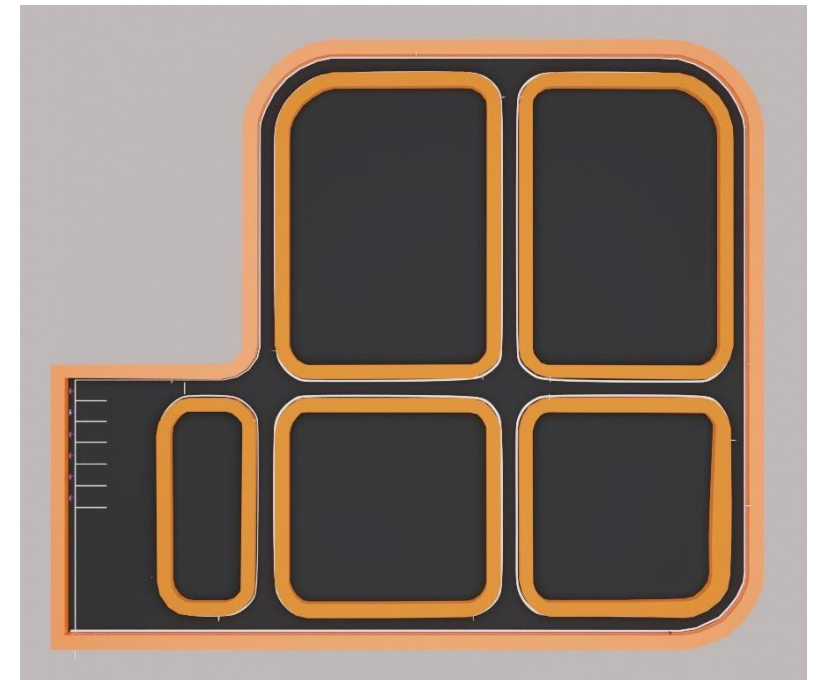


Bild 2

Simulation und Prüfung

Die erstellte Wettbewerbsstrecke wurde in das Programm Unreal Engine 4 übertragen.

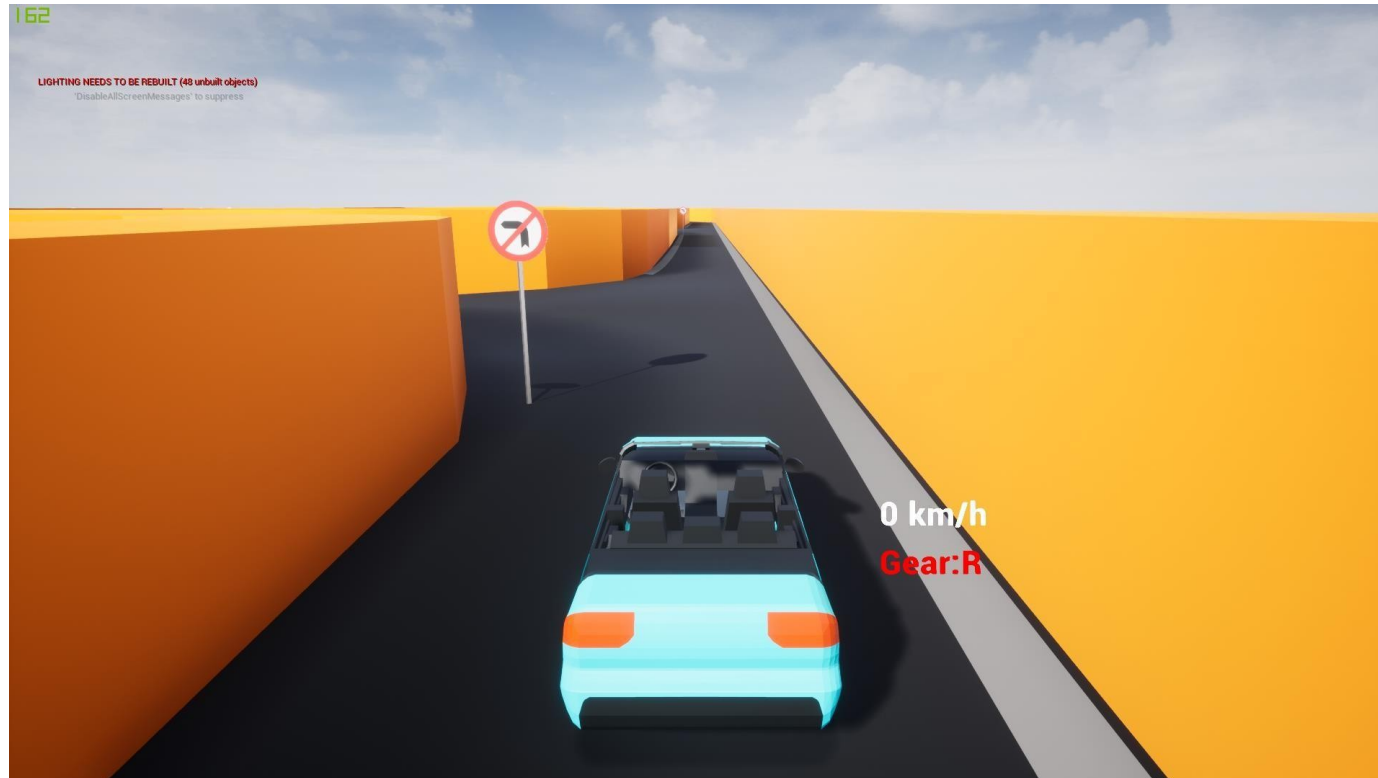


Bild 3

Juli	August	September	Oktober	November	Dezember	Januar	Februar	März	April	Mai	Juni	Juli	Arbeitspakete und Aktivitäten
													Konzept und Entwurf
													Literaturübersicht
													Konzeptueller Entwurf
													Entwurf eines Designs
													Simulation
													Detaillierter Entwurf und Materialauswahl
													Detaillierter Entwurf
													Materialauswahl
													Entwurf Erstellung
													Prototypenherstellung und Materialmontage
													Herstellung und Montage von Prototypen
													Elektrizität Elektronik Montage
													Kamera, Sensor- und Lidar-Installation
													Prototyp und Systemtests
													Fahrzeug-Fahrwerkstest



									1			Sensor Test
									2			Test zum autonomen Fahren
									1			Autonomes Parken Test
									1			Kommunikationstest

Budget

Wir haben ungefähr 17000 TL gekostet und wir haben keine Budgetanfrage von Ihnen.



- <https://www.blender.org/>
- <https://pjreddie.com/darknet/yolo/>
- <https://www.youtube.com/watch?v=hRzBtQh6wuU&list=PLzcys7whQ6eSJdSHJh-xdOLvWPP24J76x>
- https://www.youtube.com/watch?v=MJUJ4wbFm_A&list=LLMY8FCvso79ap5FnE-JhBuA&index=143
- https://www.youtube.com/watch?v=SWYqp7iY_Tc&list=LLMY8FCvso79ap5FnE-JhBuA&index=144
- https://en.wikipedia.org/wiki/CAN_bus
- <https://arxiv.org/pdf/1804.02767.pdf>
- <https://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Trafik/TehlikeUyarilsaretleri.aspx>

- J. Redmon, A. Farhadi, "Yolov3: An Incremental Improvement", arXiv:1804.02767, 2018.
- V. Gaikwad und S. Lokhande, "Lane departure identification for advanced driver assistance", IEEE Trans. Intell. Transp. Syst., vol. 16, Nr. 2, S. 910-918, 2015.