

## AVA – 2. Satus Update

### Schon Ausgeführte Arbeiten:

- **Projektplanung:**

Für die Projektplanung wurde die Gruppen eingeteilt, und ein GitHub Repository angelegt. Ebenfalls wurde der Name AVA (Automatische Verkehrs Auszählung) gewählt.

Die Gruppenmitglieder sind: *Sebastian Toldo, Paul Schickelgruber, Boris Ballada und Thomas Spindler*

Link zum Github Repository: [https://github.com/MctomSpdo/Raspberry\\_Fahrzeugzaehlung](https://github.com/MctomSpdo/Raspberry_Fahrzeugzaehlung)

Das GitHub Repository läuft über Thomas Spindler

- **Aufsetzen des Raspberry pies:**

Der Raspberry Pie (Raspberry 4B 8GB) wurde mit dem Raspberry PI OS aufgesetzt. Ebenfalls wurde OpenVC für Python installiert.

Um die Verwendung des Pies zu vereinfachen, und es ebenfalls möglich zu machen, von zuhause aus zu arbeiten, wurde auf dem Raspberry eine RDM (xrdp) und eine SSH Verbindung eingerichtet.

- **Webserver:**

Es wurde ein neuer Webserver mithilfe von NodeJS eingerichtet. Dieser verwendet aktuell nur das Module Express, und eine statische Website zu hosten.

Der Webserver ist temporär in der Entwicklung pausiert, bis die Entwicklung der Website oder OpenCV diesen benötigt.

- **Entwicklung der Website:**

Die Anfänge der Website, die dann später die Daten ausgeben sollte, wurde entwickelt. Dieser Schritt der Entwicklung ist noch immer aktiv und wird von Sebastian und Paul ausgeführt. Aktuell wird daran gearbeitet, Daten mithilfe von ChartJS anzuzeigen. Diese werden dann später von dem Webserver abgefragt.

- **Python & OpenCV:**

Es wurde Python auf dem Raspberry vorbeireitet, und auf diesen dann mithilfe von OpenCV zu entwickeln. Die ersten Tests mit der Raspberry pi camera und der Gesichtserkennung wurden ausgeführt. Momentan wird ein Modul entwickelt, das Bewegungen im Bild feststellt und diese in einem Array von Vektoren speichert. An diesem arbeitet momentan Thomas Spindler.

## Arbeitsaufteilung:

NAME	ARBEIT
Ballada Boris	Keine Arbeit aufgrund fehlender Anwesenheit / Kommunikation
Schickelgruber Paul	Webserver + Website
Spindler Thomas	Raspberry + OpenCV + Verwaltung GitHub
Toldo Sebastian	Webserver + Website

Die Arbeitsteilung wurde anhand von Verständnis der Technologien aufgeteilt.

## Deatilierte Reports:

### Webserver + Website:

#### Webserver:

Der Webserver wurde auf NodeJS mit der Verwendung von Express aufgesetzt. Dieser dient momentan als statischer Fileserver, und hat bereits 2 einfach commands, um die GET Protokolle um Kommunikation zu testen.

#### Website:

Auf der Website wird momentan daran gearbeitet, dass die Zahlen, die später von der Zählung kommen, in einem schönen und dynamischen Weg darzustellen. Für dies wurde ChartJS verwendet, da dies ein einfacher, guter, und schöner weg ist, die Zahlen auf Charts darzustellen.

### Raspberry Pi:

#### Setup:

Auf dem Raspberry Pi wurde Raspberry PI OS desktop installiert. Ebenfalls wurden Remoteverbindungen eingerichtet:

- SSH (shh) Port 22
- Remote desktop connection (xrdp) Port 3389

Ebenfalls wurde anfangs mit Docker gearbeitet, dies wurde dann allerdings auf Empfehlung der Lehrer eingestellt.

## **NodeJS**

Auf dem Raspberry wurden ebenfalls NodeJS und npm (Node package manager) installiert, um anfängliche Verbindungen zu testen.

Ebenfalls wurde Python und das pip package OpenVC (v2) installiert.

## **Python:**

Es wurde sich mit Python und der Programmier-art von dieser Sprache bekannt gemacht. Ebenfalls wurden die Testübungen ausgeführt.

Des Weiteren wurde die Verbindung zwischen Python und NodeJS hergestellt und getestet, damit später daten problemlos übertragen werden können.

Gerade wird an der Erkennung von Bewegung auf dem Bild gearbeitet. Diese soll mithilfe von Unterschieden in dem Bild passieren. Diese wird dann in Vektoren gespeichert, mit denen dann in den nächsten Schritten weitere Berechnungen ausgeführt werden. (Bild Im Anhang).

Aktuell scheitert es an Problemen mit der Kamera (oft kommt einfach kein Bild)



Herausrechnen und markieren der Unterschiede auf dem Bild

## **StatusUpdate 17.06.2021:**

- **Webserver weiterentwickelt:**  
Arbeiten an ChartJS
- **Arbeiten an OpenCV:**  
Die Unterschiede auf dem Bild in Vektoren bringen und diese grafisch ausgeben.