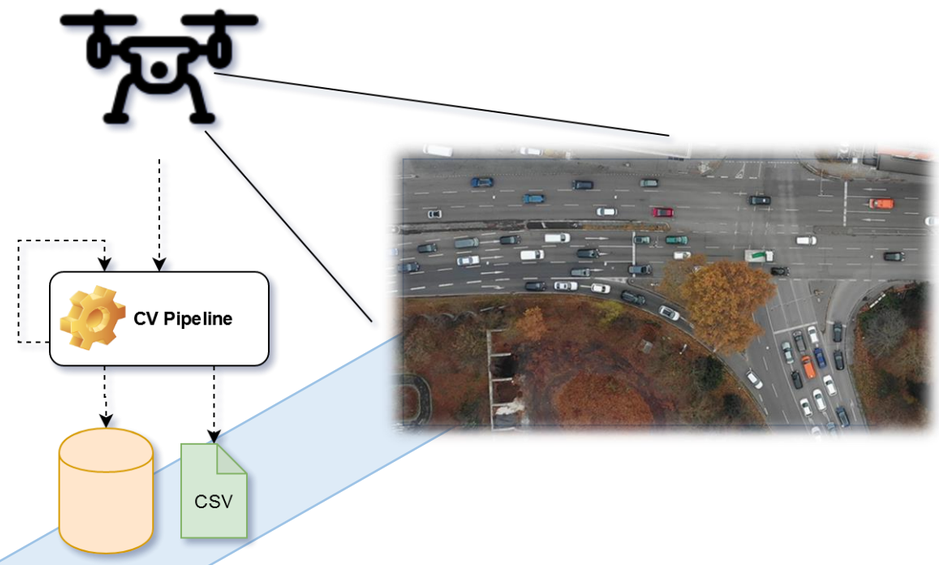
Supervisor: Dr. Paul Kyberd
Student: Michael Watzko
Student ID: 1841795

Conception and realization of a distributed and automated computer vision pipeline

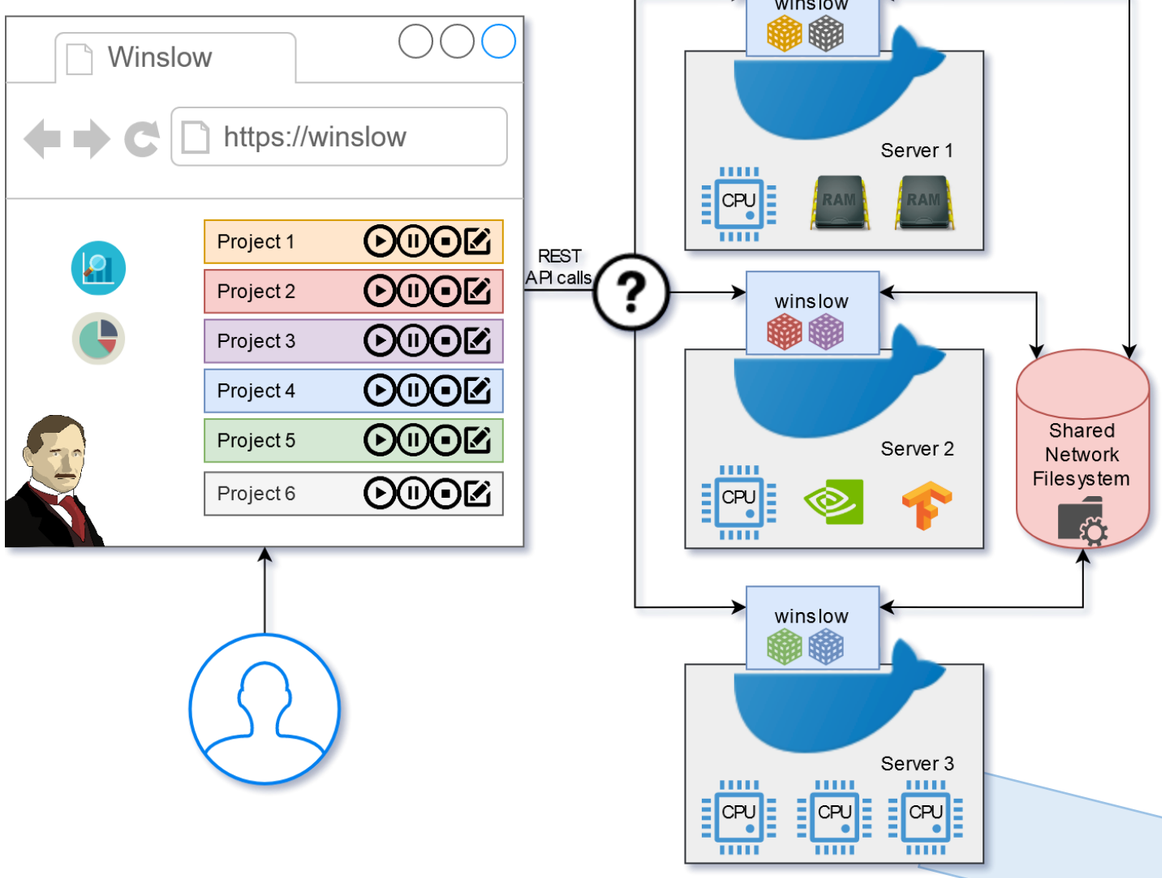
Project Context

- Detecting vehicles in video footage using Computer Vision and Artificial Intelligence
- Tracking vehicles throughout the video to determine speed, size, acceleration, class, position and lane changes
- Export data for further traffic flow analysis (in other projects or for the customer)



Main Goal

- Automate manual workflow that distributes the workload onto servers and collects the data



Further Requirements and Objectives

- Handle large files (4k video footage) and multiple projects
- Representation as multi-stage pipeline that can be paused at any stage and investigated, to re-do stages with optimized parameters
- Consider specific hardware requirements for CV and AI for each pipeline stage

Architecture, Design and Technologies

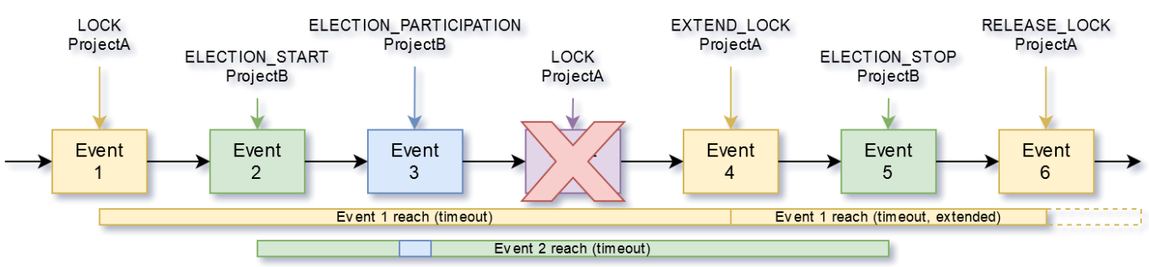
- Decentralized decision making
- Resilient against node failures
- Shared network filesystem
- Docker

Challenges and Experimental Work

- Finding a fitting network filesystem
- Communication and coordination
- Finding the most fitting execution node for a job

Results

- Time savings because of higher hardware utilization due to automatic stage execution
- Creation of a distributed and synchronous EventSystem with timeout based mutex on top



Project Progress

Task	Progress	2019 Sept	Oct	Nov	Dec	2020 Jan	Feb	Mar
Research	**DONE**							
Experimental work	**DONE**							
synchronization, coordination and communication								
managing docker container								
Implementation	**FINALIZING (99%)**							
Job distribution (algorithm)								
Error resilience on job failures, node failures and timeouts								
reacting to User-Feedback								
Metrics, Analysis and Evaluation	**60%**							
finding valuable metrics								
collect and analyse								
Thesis	**70%**							
writing everything down								

Challenges and Experimental Work

- Finding a fitting network filesystem
  - Some require big installation overhead
  - Truly decentralised filesystems are rare
  - Tons of different centralized network filesystems
  - Some provide site awareness or replication services
- How to manage separate execution history from project and pipeline template
- Can you use a shared filesystem for communication and coordination to strip down external (system) dependencies

Results

- Synchronous EventSystem with Boradcast functionality based on files on a shared filesystem
- Implementations of a timeout Mutex on-top of the EventSystem to lock projects throughout the whole system

Further Requirements and Objectives

- Automatically distribute jobs onto computing nodes
- Handle large files (4k video footage) and multiple projects
- Representation as multi-stage pipeline that can be paused at any stage and investigated, to re-do stages with optimized parameters
- Consider specific hardware requirements for CV and AI

Architecture, Design and Technologies

- Decentralized decision making
- Resilient against node failures
- Shared network filesystem for data, configuration and coordination
- Docker for easy installation of additional compute nodes

Challenges and Experimental Work

- Finding a fitting network filesystem
- Solely depend on a shared filesystem for communication and coordination to strip down external (system) dependencies

Results

- Synchronous EventSystem with Boradcast functionality based on files on a shared filesystem
- Implementation of a timeout based mutex on-top of the EventSystem to lock projects throughout the whole system
- Much more time efficient because of higher hardware utilization due to automatic stage execution