

# MICHAEL SCHLOSSER

## Software Engineer

mschlosser.se@gmail.com | +49 157 388 38 368 | Eschenweg 1, 73540 Heubach

### KURZVITA

Software Engineer mit Fokus auf KI-gestützte Bildverarbeitung, Embedded Systems und Python/C++-Entwicklung. Erfahrung in Konzeption, Implementierung und Integration von Software auf unterschiedlichen Hardwareplattformen sowie in Datenaufbereitung, Analyse und Dokumentation. Vertraut mit agiler Softwareentwicklung und interdisziplinärer Teamarbeit.



### SKILLS

**Programmiersprachen:** C++, Python, Java, Bash, Javascript

**Frameworks & Libraries:** ROS2, PyTorch, OpenCV, Open3D, OpenMMLab, TensorRT, GStreamer, Qt

**Tools & Plattformen:** Linux, Git, Docker, CMake, Buildroot, Android, GoogleTest, PyTest, VS Code, Atlassian Suite (Jira, Confluence, Bitbucket)

**Konzepte:** (3D) Computer Vision, Deep Learning, Agile Development, Test-Driven Development, Software Architecture & Design Patterns, Parallel Programming & Concurrency, Embedded Systems

**Sprachen:** Deutsch (Muttersprache), Englisch (verhandlungssicher), Französisch (gute Kenntnisse), Russisch (Grundkenntnisse)

### BILDUNGSWEG

#### Master of Science – Robotics, Cognition and Intelligence

10/2022 – 09/2025

Technische Universität München

#### Bachelor of Science – Informatik / Software Engineering

10/2018 – 09/2022

Hochschule Aalen

### BERUFSFAHRUNG

#### Master Thesis

11/2024 – 06/2025

TUM Institute of Automotive Technology, Garching

- Titel der Arbeit: "Multimodal Pseudo-Labeling for 3D Object Detection in Autonomous Driving". Note: 1.0
- Entwicklung eines Frameworks zur Sensorfusion von LiDAR- und Kameradaten zur Verbesserung der 3D Pseudo-Labeling Pipeline
- **Verwendete Technologien:** Python, PyTorch, OpenMMLab, OpenCV, Open3D, NumPy, Docker, Git, LaTeX, VS Code

#### Werkstudent – Perception & Sensor Systems

10/2022 – 10/2024

HAT.tec GmbH, Neubiberg

- Entwicklung von ROS2-Anwendungen für die Verarbeitung und Analyse verschiedener Sensordaten in einem agilen Team
- Training, Evaluierung und Optimierung eines Objekterkennungsmodells für Luftfahrzeuge für verschiedene Zielplattformen
- Annotation und Analyse von firmeninternen Datensätzen
- **Verwendete Technologien:** C++, Python, ROS2, Qt, PyTorch, TensorRT, OpenMMLab, OpenCV, CMake, GoogleTest, Linux, Docker, Git

#### Praktikant / Werkstudent / Bachelor Thesis – Computer Vision & Image Processing

09/2020 – 09/2022

HENSOLDT Optronics GmbH, Oberkochen

- Bachelor Thesis: "Evaluation und Optimierung von Deep-Learning basierter Objektdetektion auf eingebetteter Inferenz-Hardware von Xilinx und Nvidia". Note: 1.3
- Veröffentlichung: [A Framework for Benchmarking Real-Time Embedded Object Detection, DAGM GCPR 2022](#)
- Setup einer Object-Detection-Demo mit TensorRT, Deepstream und Vitis AI
- Aufbau einer Buildroot-Linux-Umgebung für Embedded-Deployment auf Zynq UltraScale+
- Implementierung hardwarebeschleunigter Bildverarbeitung auf FPGAs mit Xilinx Vitis Vision.
- **Verwendete Technologien:** C++, CMake, Buildroot, TensorRT, Deepstream, OpenVino, GStreamer, Python, Qt, Linux, Xilinx Vitis Vision Libraries, Xilinx Vitis AI, Vivado, Git, Docker, Confluence

### PROJEKTE

#### Face Touching Detector

[github.com/mischlox/face-touching-detector](https://github.com/mischlox/face-touching-detector)

- Desktop-App zur Echtzeit-Erkennung von Gesicht-Berührungen
- YOLOv5m (Libtorch C++), QT5-GUI, Docker-Deployment
- **Verwendete Technologien:** C++, Libtorch, Qt, Docker, CUDA

#### Augmented Reality Object Recognition App

[github.com/mischlox/AR-App-Object-Recognition](https://github.com/mischlox/AR-App-Object-Recognition)

- Android-App zur Echtzeit-Objekterkennung und -einlernung mit Smartphone-Kamera
- Continual Learning zur Erweiterung eines MobilenetV2-Modells direkt auf dem Gerät, Offline-fähig
- **Verwendete Technologien:** Java, TensorFlow Lite, Android SDK

#### Mitarbeiter-Zeiterfassungssystem

- Desktop-Tool mit Login, Zeiterfassung, Admin-Dashboard, Soll-/Ist-Stunden, Urlaubs- & Krankheitsverwaltung, PDF-Export
- Produktiv in 3 Supermarkt-Filialen
- **Verwendete Technologien:** Python, SQLite, JavaScript, Eel, HTML, CSS