

Alexander Epple

AI Master's Student · Computer Vision & Applied Machine Learning

Geltendorf, Germany | alexander.epple@uni-ulm.de | 0152 06182053 | alexanderepple.de | github.com/dralois

Master's student in Artificial Intelligence at Ulm University specializing in computer vision, deep learning, and applied machine learning, with 5+ years of professional software engineering experience. I build robust, well-structured systems — from self-supervised vision pipelines in PyTorch to a GPU-accelerated rendering engine that replaced a slow legacy drawing path in production software. Currently seeking a working-student position, internship, or master's thesis in computer vision / machine learning.

PUBLICATIONS

A. Epple *et al.*, “Bar-JEPA: Extracting Values from Bar Charts with a Joint-Embedding Predictive Architecture,” accepted at the 20th International Conference on Document Analysis and Recognition (**ICDAR 2026**), Vienna — main conference track. *First author.*

EDUCATION

M.Sc. Artificial Intelligence — Ulm University Apr 2025 – Feb 2027 (expected)

Grade average ≈ 1.2 (current)

M.Sc. Games Engineering — Technical University of Munich (transferred to M.Sc. AI, Ulm) Oct 2020 – Sep 2022

B.Sc. Games Engineering — Technical University of Munich 2014 – 2020

Thesis (graded 1.0): photorealistic synthetic training data for object detection and 6D pose estimation. (Studies paused 2015–2017 for a full-time engineering role.)

Abitur (High School Diploma) — Ignaz-Kögler-Gymnasium 2014

EXPERIENCE

Independent Projects — Embedded / IoT & Hardware Prototyping 2022 – 2025

- Designed, built, and programmed **ESP8266-based IoT devices**, including a beverage-fermentation/keg monitor and a wallbox power-automation controller.
- Self-taught embedded electronics and 3D-printed hardware design; iterated on custom devices end-to-end and maintained a personal portfolio site.

Working Student / Software Engineer — ATX Hardware GmbH 2017 – 2020

- Replaced the NH-Software legacy GDI+ software-rendering path with a **GPU-accelerated 2D engine** (modified SFML, C#/C++, custom shaders), eliminating multi-second UI freezes and keeping large circuit-board projects responsive in daily use.
- Added Line-Arc DXF geometry support to the board-analysis pipeline and built a marching-squares **outline-extraction algorithm** that converts safety-overlay regions into precise line/arc vector geometry.
- Built control software for the “Lomecs” laser-guided optomechanical imaging machine (VB.NET, serial motion control) and a Raspberry-Pi camera system.

Software Engineer (full-time) — ATX Hardware GmbH 2015 – 2017

- Developed and maintained in-house .NET software for circuit-board test-adaptor design, used daily by the sales and engineering teams.

SELECTED PROJECTS — COMPUTER VISION & MACHINE LEARNING

- Bar-JEPA** — self-supervised value extraction from bar charts. Fine-tuned an **I-JEPA** encoder, added variable-resolution ViT inputs, and generated a 100k synthetic-chart dataset; a lightweight decoder recovers per-bar values. Python/PyTorch, Weights & Biases.
- Vision-Language-Action Models** — seminar paper (Visual Computing, Ulm, 2026) on visual **Chain-of-Thought** reasoning for robotic VLAs — autoregressive sub-goal generation, action tokenization, and world-model directions.
- B.Sc. Thesis & Guided Research** — photorealistic synthetic training data via a physics engine (Blender, NVIDIA PhysX, C++/embedded Python); trained Mask R-CNN on 10k generated images. Extended with HDR light-source estimation.
- WorldModel²** — model-based RL (**DreamerV3**) for transferable exploration in procedurally generated environments. Built a custom rule generator with a Z3 ground-truth solver and a GPU-accelerated simulator; Dreamer variants beat all model-free baselines on hard rulesets.
- LangueViz** — spoken-language identification from mel-spectrograms via **DenseNet** transfer learning, with large-scale W&B sweeps — a practical case study in dataset bias and distribution shift.
- ParkDas** — multimodal Telegram parking assistant with LLM intent parsing (Google Gemini) and map-based output. Led backend development and AI integration.

SKILLS

Research interests computer vision, self-supervised learning, vision-language-action & generative models

Programming Python, PyTorch, C# / .NET (C#/VB), C / C++, HLSL/GLSL

Familiar with NumPy / Pandas, scikit-learn, Hugging Face Transformers

ML / CV deep learning, neural networks, CNNs & Vision Transformers, model-based reinforcement learning, synthetic data generation, experiment tracking (Weights & Biases)

Tools & Systems Linux, Slurm, Docker, Git, embedded (ESP8266, Raspberry Pi, microcontrollers), Blender

Languages German (native), English (C2, near-native)

Beyond coursework MIT 6.S184: Flow Matching & Diffusion Models — self-study labs (github.com/dralois/FM-Labs); weekly ML paper-reading group with PhD students.