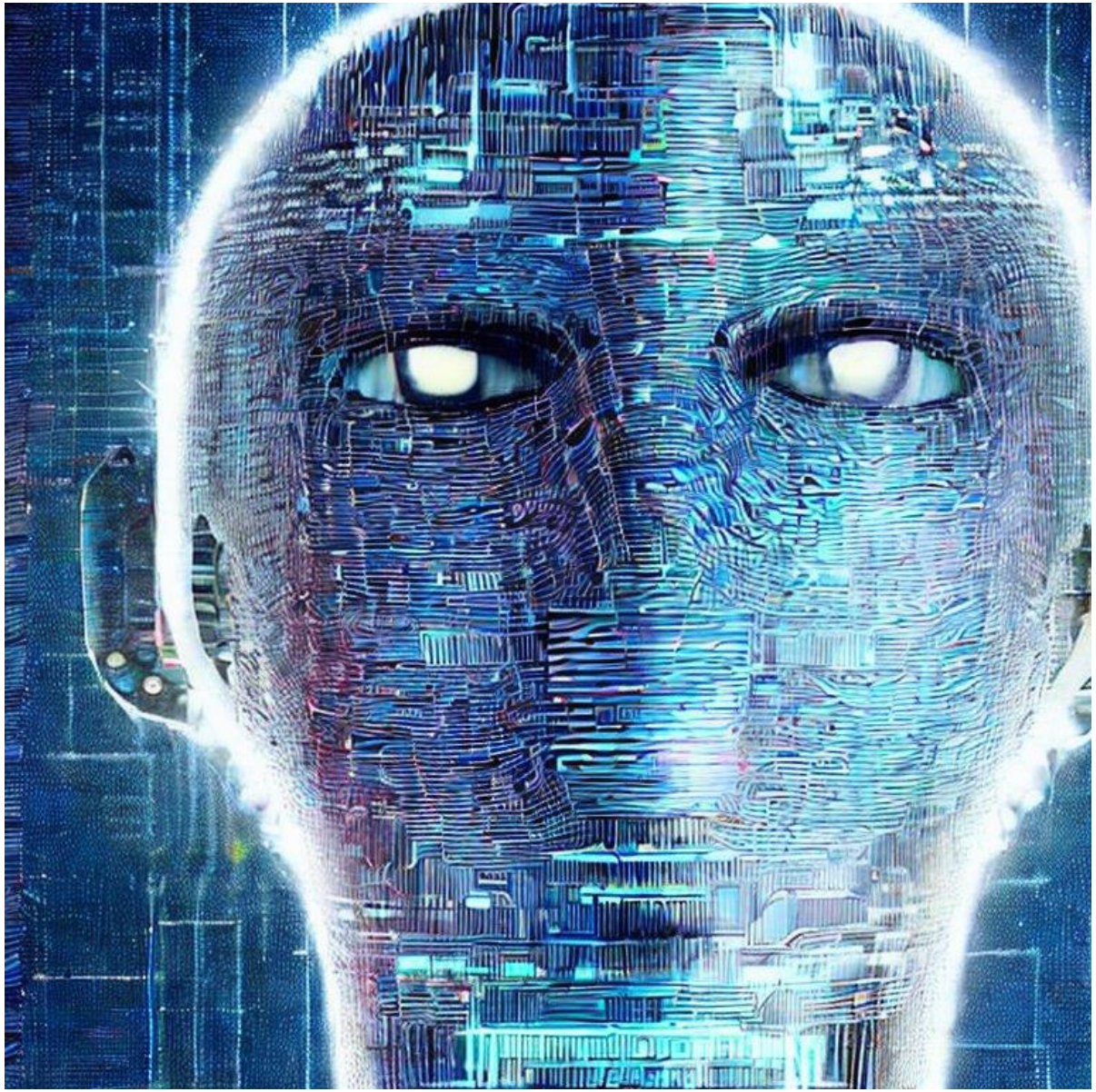


computer vision on edge devices

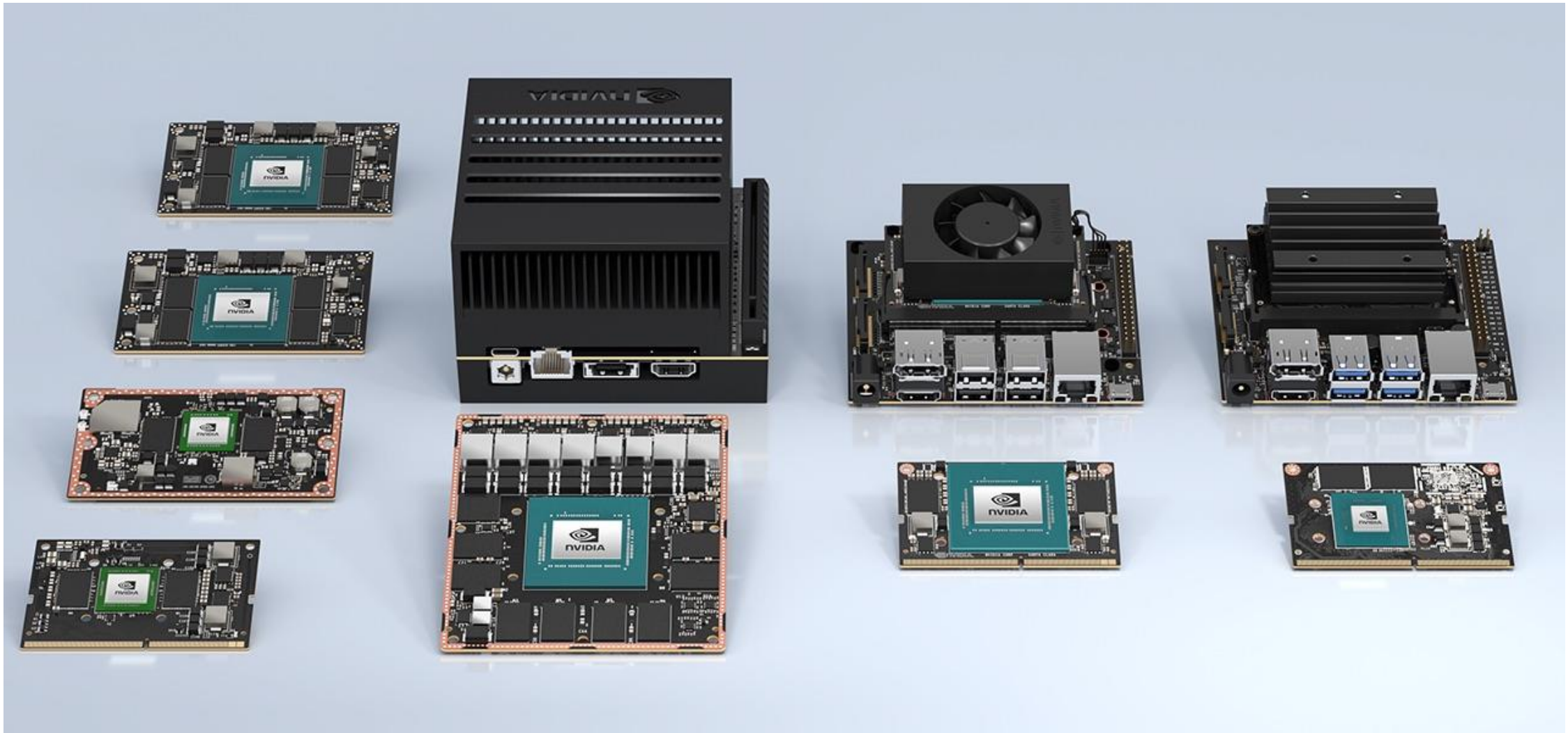
TONI BADERTSCHER

MACHINE LEARNING PROJEKT

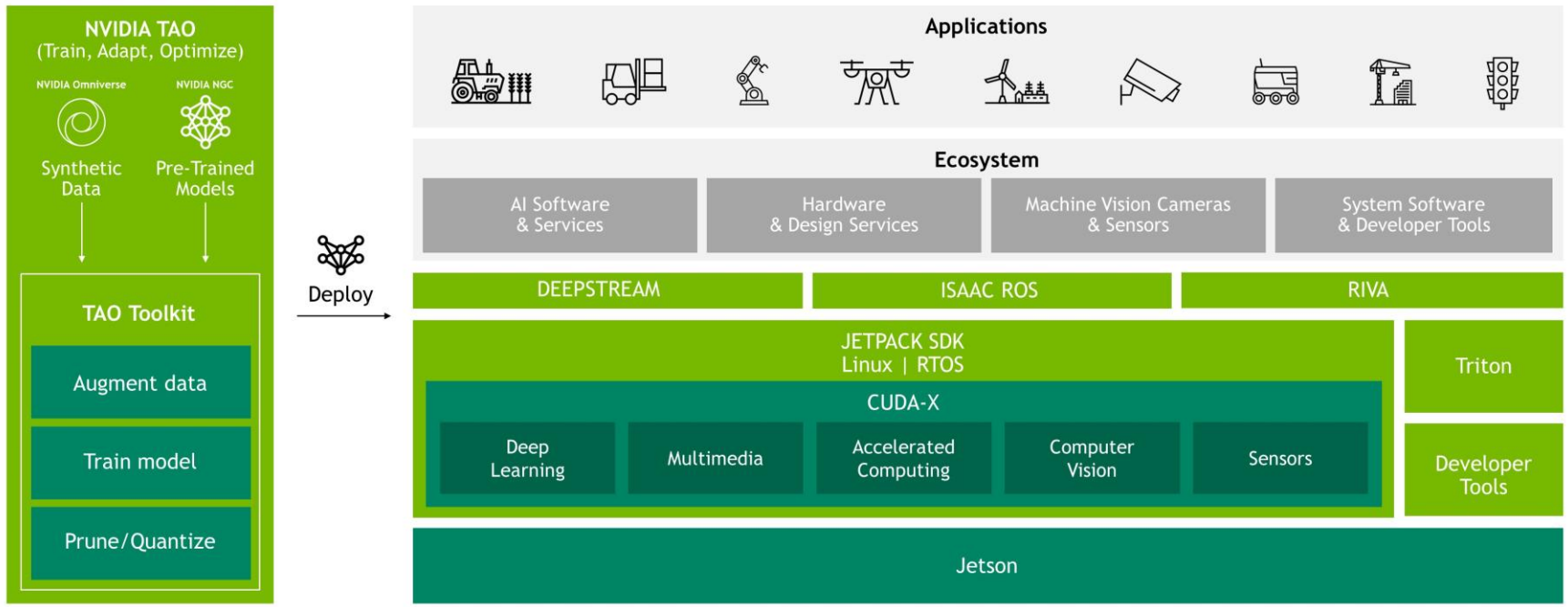
Agenda

- Projektidee / Motivation
- Timeline
- Hardware setup, Entwicklungsumgebung
- Analyse, funktionale und nicht-funktionale Anforderungen
- MVP
- Erweiterungen, Verbesserungen
- Fazit, lessons learned

Jetson hardware & dev kits


NVIDIA

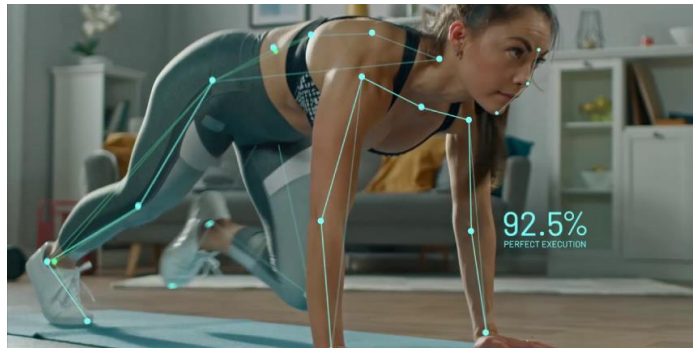
JETSON SOFTWARE

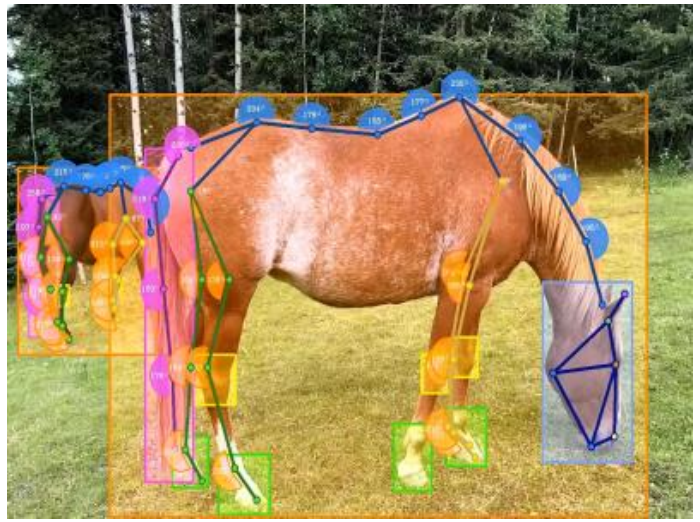


pose estimation / classification

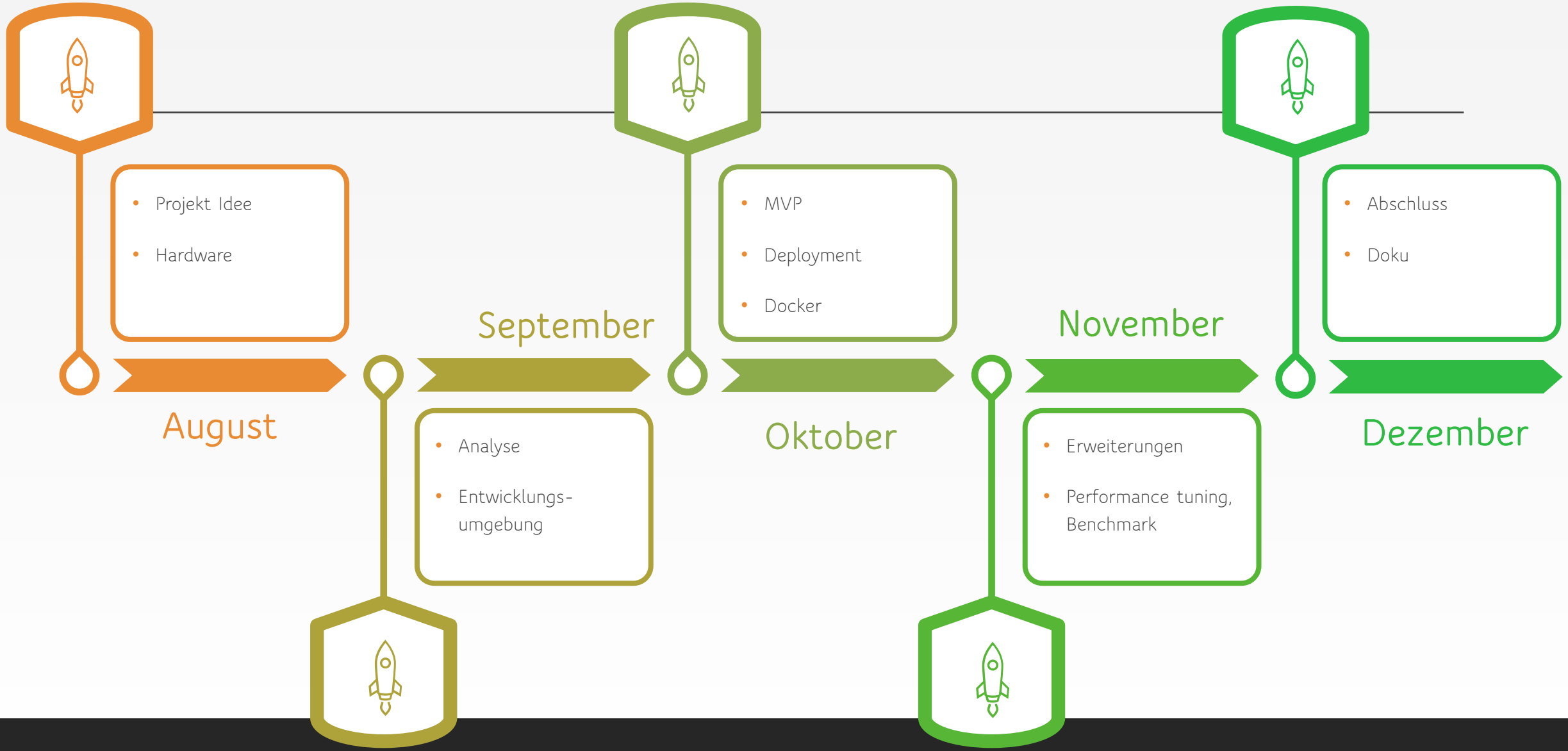
Anwendungsbereiche

- human - robot interaction
- fitness tracker
- motion tracking for gaming
- avatar / metaverse
- livestock monitoring



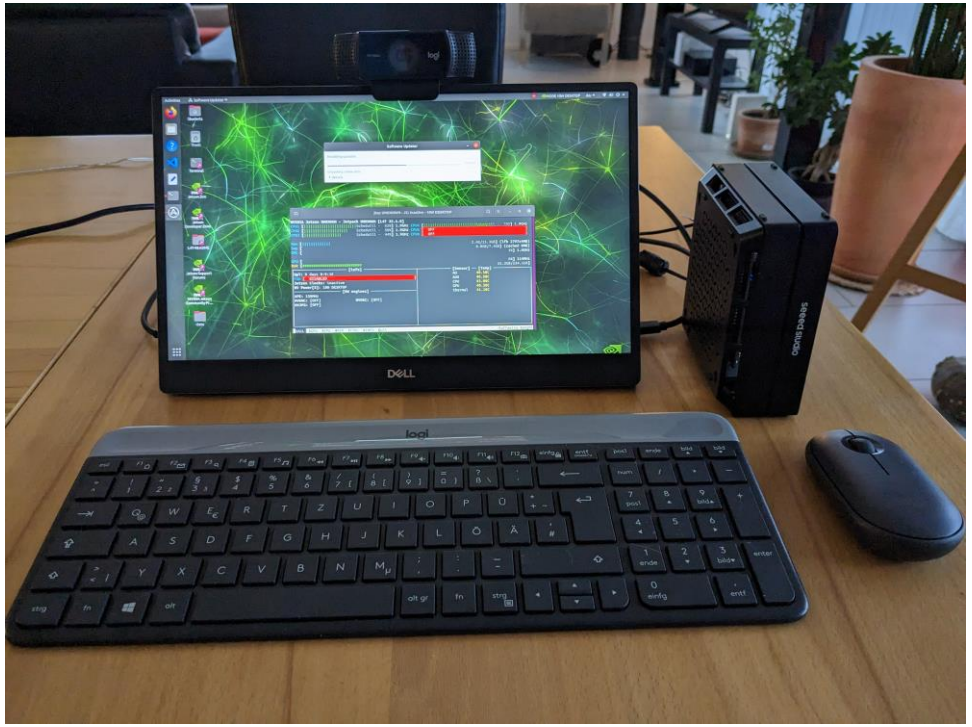


Projekt timeline

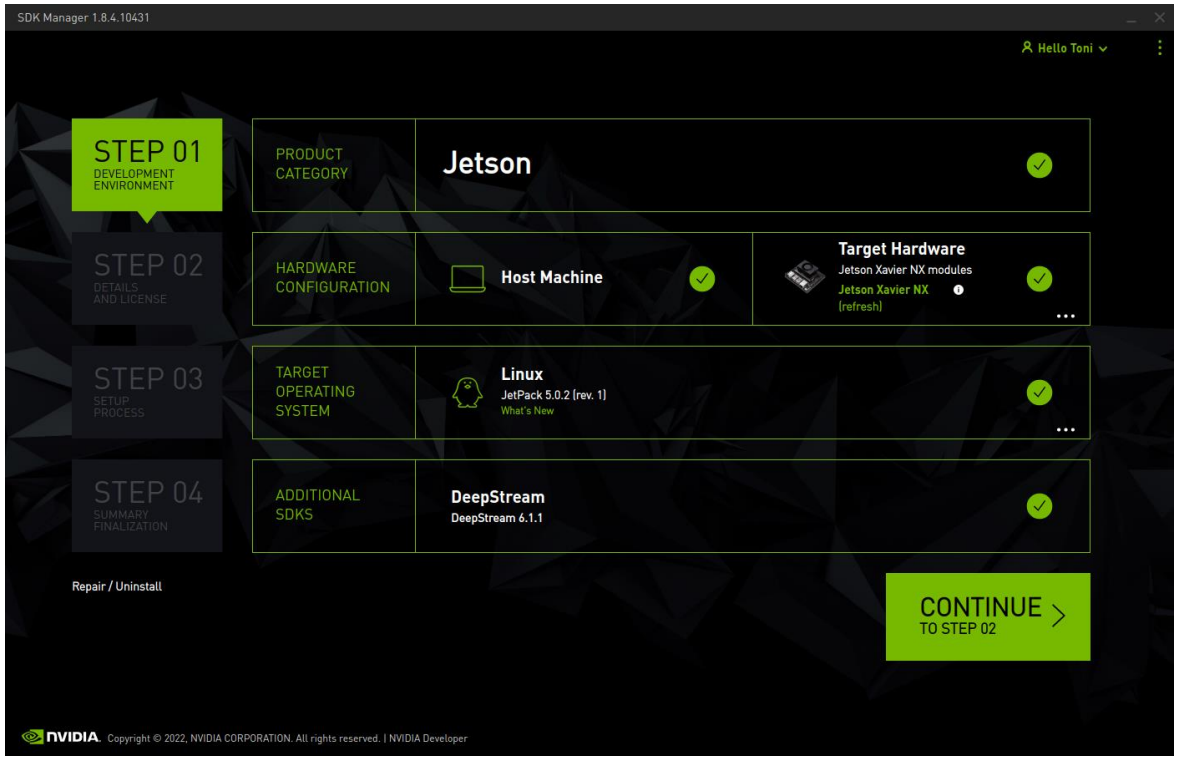


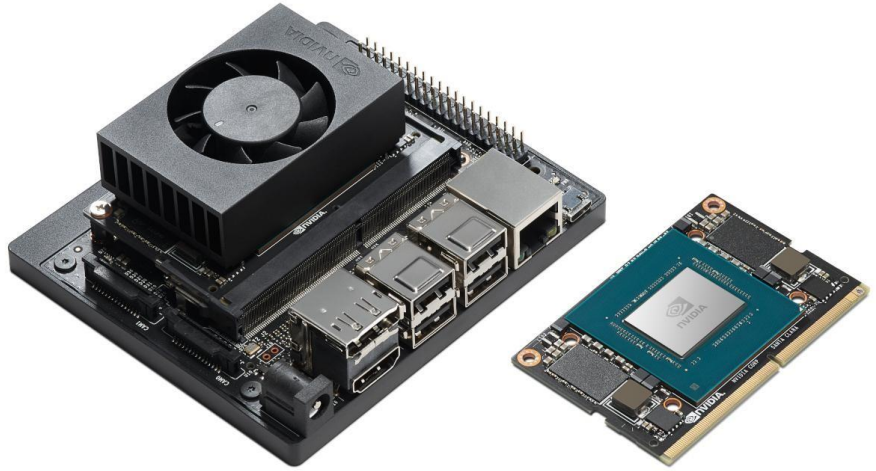
Hardware setup, tools

- Jetson Xavier NX, 16GB RAM, 256GB SSD, USB Webcam, RTSP camera
- Jetpack 5.02
- Ubuntu 20.04, ARM
- Cuda, cuDNN, TensorRT, Pytorch, Deepstream, TAO
- Docker, Nvidia NGC CLI
- X86 Laptop, 32GB RAM, RTX3060 6GB, Ubuntu 20.04



Jetpack, SDK Manager





Analyse

- Olga Chernytska, 2D Hand Pose Estimation-RGB, FreiHand dataset
- NVIDIA AI IOT: trt_pose, trt_pose_hand
- Google mediapipe (python)
- MMPose (OpenMMLab Pose Estimation Toolbox)
- YOLOv7
- Nvidia TAO, DeepStream, Triton

Erkenntnisse aus der Analyse

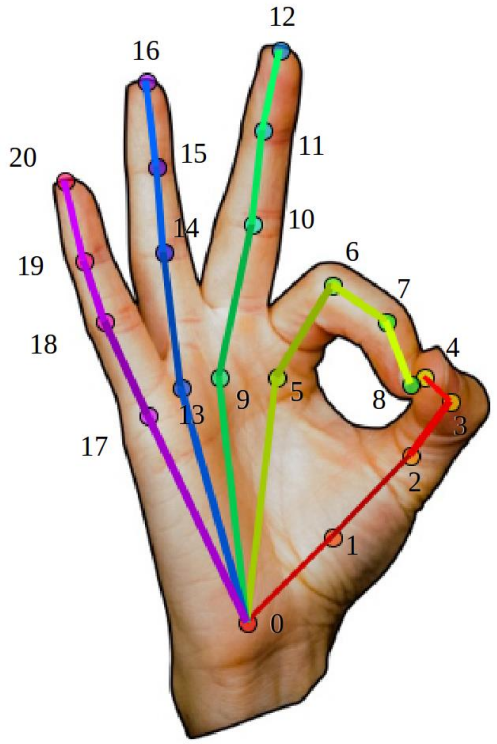
- 2D order 3D basierte Ansätze
- top-down oder bottom-up Modellarchitektur
- Ingesamt recht hohe Performanceanforderungen bei vielen Modellen
- Unterschiedliche keypoint Formate bei human pose
- Viele Modelle haben Probleme in speziellen Anwendungsfällen (z.B. Yoga)
- Erstellen eines Datensets für Classifier könnte aufwendig sein (speziell bei Yoga Posen)

Anforderungen

- 2d oder 3d keypoint estimation auf der Basis von RGB input (webcam)
- Klassifizierung von Handzeichen
- single person (ein oder beidhändig)
- als Basis sollen PyTorch Modelle dienen, Erweiterbarkeit
- real-time, mindestens 5fps
- target platform: X86 und jetson (ARM)

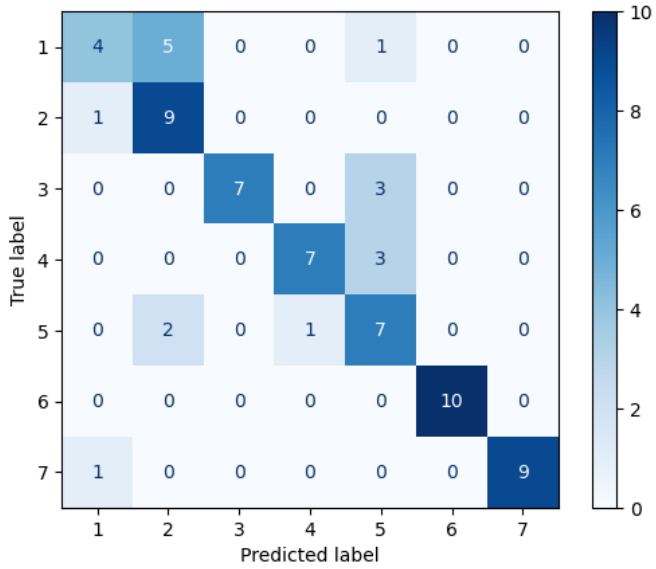
MVP

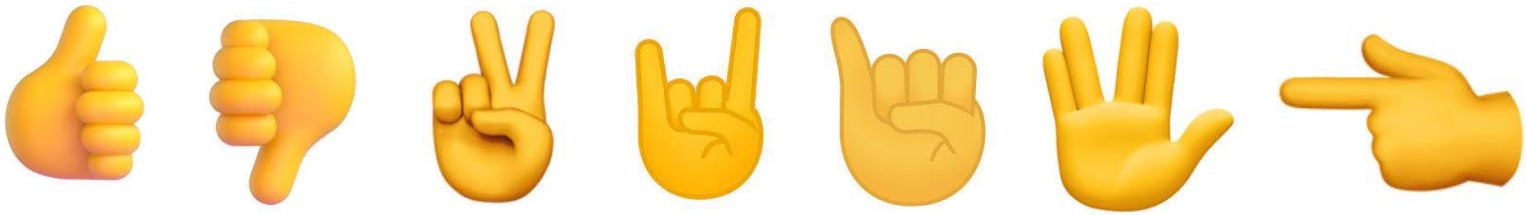
- basiert auf NVIDIA IOT demo (trt_pose / trt_pose_hand)
- resnet18 baseline
- single hand, 2d keypoint estimation
- scikit-learn SVM classifier
- torch2trt (PyTorch to TensorRT converter)



Classifier

- Datenset mit 7 Klassen (home made)
- 40 Bilder pro Klasse für Training
- 10 Bilder pro Klasse für Test
- svm accuracy: 0.76





Performance (Cuda, TensortRT)

- fp16 half-precision inference performance
- Xavier NX, 12 nm, 384-core NVIDIA Volta GPU, 48 Tensor Cores
- RTX 3060 notebook GPU, 8 nm, 3840-core Ampere GPU

	batch size 1	**batch size 8**
AMD Ryzen RTX3060 **CPU**	118.93 ms	714.60 ms
AMD Ryzen RTX3060 **GPU**	5.43 ms	30.56 ms
AMD Ryzen RTX3060 **TRT**	1.07 ms	16.46 ms
Jetson Xavier NX **CPU**	7534.95 ms	28720.30 ms
Jetson Xavier NX **GPU**	36.37 ms	343.61 ms
Jetson Xavier NX **TRT**	5.96 ms	24.80 ms

Ertweiterungen, SOT

- MMPose: top-down Ansatz funktioniert deutlich zuverlässiger
- ViTPose: vision transformer for human pose estimation
- Facebook (Meta research) InterHand2.6M (3D)

Conclusion

- there is more than meets the eye
- hoher Aufwand für Analyse und Evaluation, learning curve bei frameworks
- library/framework Probleme kosten schnell viel Zeit
- SOT Lösungen haben hohe Anforderungen an die hardware
- der Bereich entwickelt sich immer noch sehr schnell. Es gibt alle 2-3 Jahre neue GPU Generationen. Die Leistungssteigerungen sind immer noch beachtlich (verglichen mit CPU)
- was mehr als 2 Jahre alt ist verliert schnell seine Nutzbarkeit (Abhängigkeiten von alten frameworks etc.)