

# Artificial First Responder

Alexander Wagner, Joshua Franz, Niels Schneider und Marius Gerold Fakultät Informatik, Hochschule Furtwangen  
Furtwangen, Deutschland

Email: alexander.wagner@hs-furtwangen.de, joshua.franz@hs-furtwangen.de, niels.schneider@hs-furtwangen.de,  
marius.gerold@hs-furtwangen.de

**Abstract**—In einem Unternehmen werden Evakuierungshelfer unter anderem dafür eingesetzt in einer Notfallsituation (z.B. Feueralarm) ein Gebäude schnell evakuieren zu können. Der Evakuierungshelfer überprüft bei einer Evakuierung entsprechende Räume, um sicherzugehen, dass sich keine Person mehr im Gebäude befindet. Diese Aufgabe soll der in diesem Projekt entwickelte Artificial First Responder übernehmen. Zusätzlich soll dieser klassifizieren, ob sich gefundene Personen in einer Gefahrensituation befinden.

## I. EINFÜHRUNG

Damit sich im Falle eines Notfalls, kein Ersthelfer in Gefahr bringen muss, soll die Evakuierung robotergestützt ablaufen. Dies soll im letzten Ausbauschnitt über den Roboter temi<sup>1</sup> erfolgen. Da sich auf diesem ein Android-Tablet befindet, wurde eine Flutter<sup>2</sup>-Applikation entwickelt, die Bilder an ein Backend zur Auswertung schickt. Der Schwerpunkt dieser Arbeit liegt in der Entwicklung eines Systems zur kameragestützten Personengefährdungserkennung. Das komplette System wird als Artificial First Responder (AFR) bezeichnet.

## II. ARCHITEKTUR

In Abbildung 1 sind die Microservices des AFR zu sehen welche in einem Kubernetes-Cluster<sup>3</sup> laufen. Die einzelnen Komponenten werden im Folgenden genauer erläutert. Die Cluster interne Kommunikation zwischen AFR-Controller und Image-Processor/Pose-Estimator ist über gRPC realisiert. Die externe Kommunikation zwischen AFR-Controller und der Flutter-App ist mittels des MQTT-Broker HiveMQ<sup>4</sup> umgesetzt. Die Flutter-App läuft auf einem Android-Gerät. In unserem Test-Aufbau sendet diese Bilder per Knopfdruck an den Message Broker HiveMQ.

Der AFR-Controller liest die Bilder aus dem Topic des HiveMQ und sendet sie per gRPC an den Image-Processor. Die Rückgabe des Image-Processors gibt der AFR-Controller an den Pose-Estimator weiter. In unserem Testsetup schickt der AFR-Controller die gefundenen und klassifizierten Personen über ein HiveMQ-Topic zur Visualisierung zurück an die Flutter-App. Der Image-Processor hat die Aufgabe einzelne Personen aus Bildern auszuschneiden. Dazu wird ein vor-trainiertes YOLOv4-Netz [1] genutzt. Personen in unnatürlicher Haltung, die gleichzeitig teilweise hinter Gegenständen sind, erkennt der Image-Processor nur

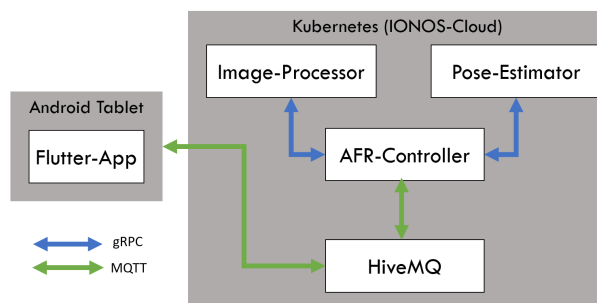


Fig. 1. Architektur: Artificial First Responder

schwer. Zur Ermittlung des Gefährdungszustandes einer Person wird die MediaPipe Pose [2] in Kombination mit Light-GBM [3] eingesetzt. Die MediaPipe Pose extrahiert Gelenkpunkte (Landmarks) aus einem Personen-Bild. Die gewonnenen Landmark-Informationen werden als Trainings-Datensatz für den "Light Gradient Boosting Machine"-Klassifikator verwendet. Für die Klassifikationen der Posen im Rahmen dieses Projekts wurde eine Genauigkeit von 89% erreicht. Das Ausschneiden der Personen mit dem Image-Processor ist notwendig, da die MediaPipe im Pose-Estimator nur mit einzelnen Personen arbeiten kann.

## III. AUSBLICK

Weiterführende Schritte für das Projekt wären das Deployment der Flutter-App auf dem temi, sowie dessen Navigation. Die Navigationsrouten des Roboters könnten im Notfall im Zusammenspiel mit dem Besuchermanagement-Tools DoorJames der CGI Inc geplant werden. Ein weiterer Aspekt wäre, dass bei einer Gefahrenklassifikation die Personen angesprochen werden. Wird im Anschluss eine Gefährdung sichergestellt, sollte eine Notfallstelle informiert werden.

## REFERENCES

- [1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [2] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.
- [3] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

<sup>1</sup>temi: <https://www.robotemi.com/>

<sup>2</sup>Flutter: <https://docs.flutter.dev/>

<sup>3</sup>Kubernetes: <https://kubernetes.io/>

<sup>4</sup>HiveMQ: <https://www.hivemq.com/>

# ARTIFICIAL FIRST RESPONDER

Alexander Wagner, Joshua Franz, Marius Gerold, Niels Schneider  
Prof. Dr. Lothar Piepmeyer

## Allgemein

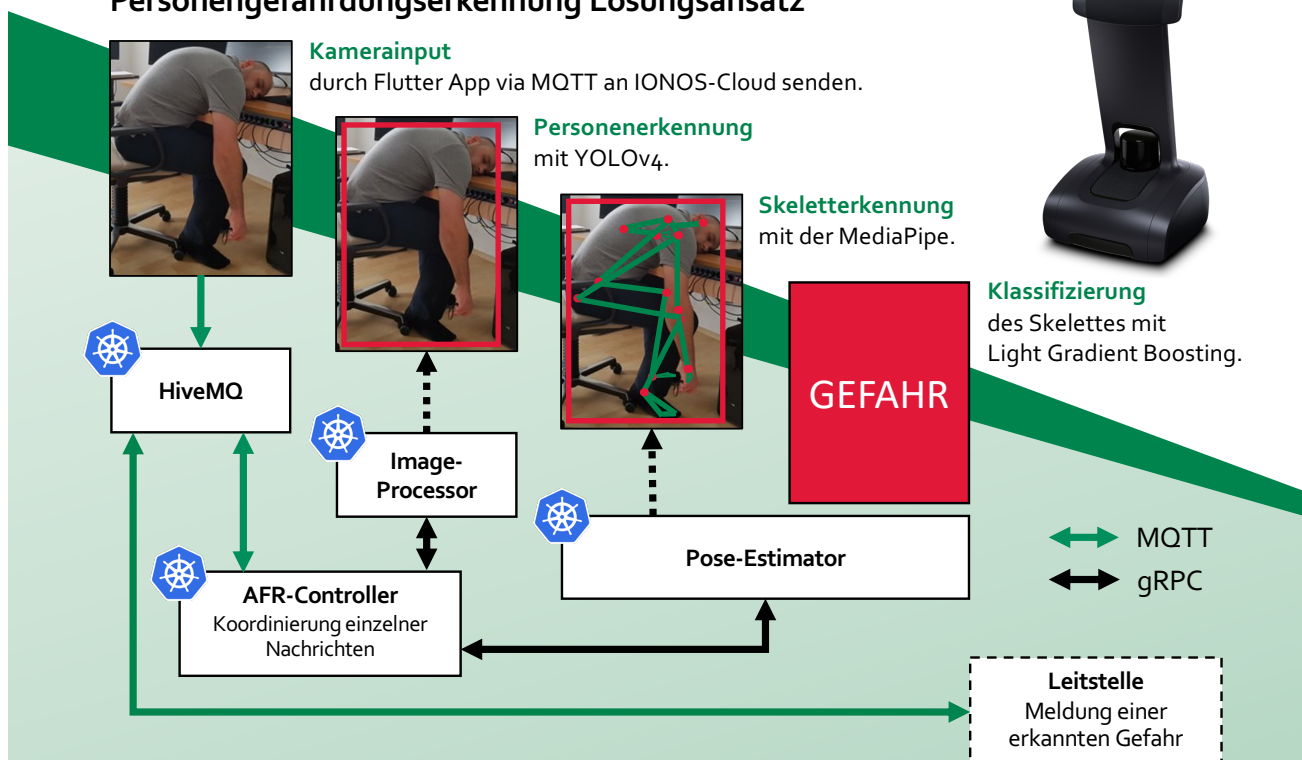
Das von CGI Inc. begleitete Forschungsprojekt **Artificial First Responder** (AFR) behandelt die Entwicklung eines roboterbasierten Evakuierungshelfers. Dieser sucht im Falle eines Notfalles (z. B. Brand) kameragestützt Personen. Mittels einer künstlichen Intelligenz (KI) bestimmt der Roboter anschließend, ob eine erkannte Person in der Lage ist, sich selbstständig zu evakuieren. Wird eine gefährdete Person erfasst, meldet der AFR dies an eine entsprechende Leitstelle.

## Anforderungen

Der Schwerpunkt des Projektes liegt auf der Entwicklung eines Systems zur Personengefährdungserkennung. Dabei soll für den sich im CGI Innovation Lab Karlsruhe befindende Assistenzroboter TEMI eine Flutter App programmiert werden.



## Personengefährdungserkennung Lösungsansatz



## Ausblick

- Installation auf dem TEMI.
- Video-Streaming aus Flutter-App.
- Integration von Door James.
- Kontaktaufnahme bei Gefährdung.