Mit Geschwindigkeit zu mehr Sicherheit

Die Indiana University – Purdue University Indianapolis (IUPUI) erforscht Möglichkeiten zur Verbesserung der Straßenverkehrssicherheit für autonome Anwendungen. Dazu untersuchen die Forscher Vorteile der Hochgeschwindigkeitsverarbeitung von Sensordaten. RTMaps Embedded und NXP BlueBox dienen als zentrale Echtzeitplattform für eingebettete Verarbeitungsfunktionen.

Entwicklung von Algorithmen für autonomes Fahren, die schnell und präzise auf die wahrgenommene Umgebung reagieren

Die Anzahl der Sensoren und Steuergeräte, die in autonomen Fahrzeugen eingesetzt werden, wächst exponentiell. Daher ist es für die Automobilhersteller entscheidend, leistungsfähige und effiziente Lösungen einzusetzen, die die Vielzahl an Aufgaben in Echtzeit ausführen. Aufgaben wie Datenerfassung, Zeitstempel-Synchronisation und die Datenverarbeitung verschiedener Sensoren wie Kameras, Laserscanner, Radar sowie von GNSS (Global Navigation Satellite System)-Empfängern müssen innerhalb von Millisekunden erledigt werden, um die genaue Position und den sicheren Betrieb des Fahrzeugs zu gewährleisten. Studenten der Purdue School of Engineering and Technology der Indiana University – Purdue University Indianapolis (IUPUI) haben mehrere Studien durchgeführt, um den Einsatz einer Berechnungsplattform zur Verarbeitung von Sensordaten für fünf verschiedene autonome Anwendungen zu testen.

Aufbau der Testplattform

Zu Beginn ihrer Forschung richteten die Studenten eine Testplattform ein und installierten die von dSPACE vertriebene Software-Lösung RTMaps Embedded (Version 4.5.0) von Intempora. Dieses Werkzeug bietet die Möglichkeit, Multisensoranwendungen in Echtzeit auszuführen, zahlreiche Software-Tasks zeitkohärent zu handhaben und Rohdatenströme mit hoher Bandbreite zu verarbeiten. RTMaps Embedded besteht aus mehreren unabhängigen Modulen: RTMaps Runtime Engine, RTMaps Studio, RTMaps Component Library und RTMaps SDK (Software Development Kit).

„RTMaps Embedded ist geeignet, um Multisensor-Herausforderungen zu meistern“, sagt Professor Mohamed El-Sharkawy, Purdue School of Engineering and Technology. „Das Tool schafft einen effizienten und intuitiven Rahmen für die schnelle und robuste Entwicklung von Fahrerassistenzsystemen, Fahrzeugen und Robotik. Wir konnten damit unkompliziert Anwendungen entwickeln, testen, validieren, vergleichen und ausführen.“ Zur Testplattform gehörte auch NXP BlueBox, eine integrierte Verarbeitungseinheit, mit der ein hochaufgelöstes 3D-Bild der Fahrzeugumgebung in Echtzeit berechnet werden kann. Im Detail verwendeten die Studenten die BlueBox Version 2.0 mit dem Visions- und Sensorfusionsprozessor S32V234, dem integrierten Kommunikationsprozessor LS2084A und dem Radarmikrocontroller S32VR27.

„NXP BlueBox bietet die erforderliche Leistung, um die erfassten Sensordaten zu analysieren sowie Algorithmen für eine zuverlässige, autonome Fahrzeugführung zu entwickeln“, führt >>

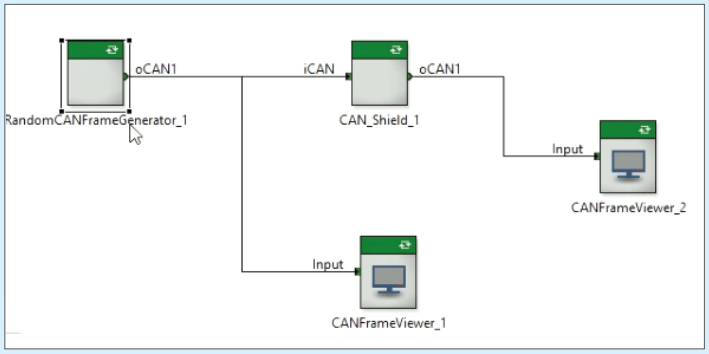


Abbildung 1: Arbeitsbereich von RTMaps für ein benutzerdefiniertes Komponentenpaket.

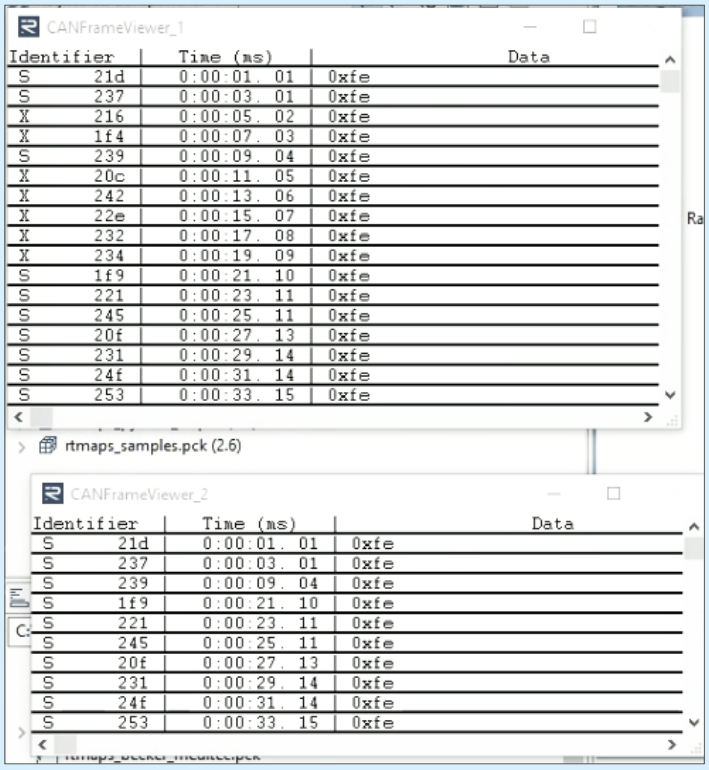


Abbildung 2: Anzeige der Ausgabe mit dem CAN Frame Viewer in RTMaps.

El-Sharkawy weiter aus. „BlueBox bietet die erforderliche Leistung, die funktionale Sicherheit und die Zuverlässigkeit, die im Automobilbereich notwendig sind, und lässt sich perfekt mit RTMaps integrieren."

Hinweis: RTMaps Embedded ist mit der dSPACE MicroAutoBox Embedded SPU kompatibel – einer Prototyping-Plattform für Funktionen für das automatisierte Fahren.

Testen der fünf Anwendungen

Mit der Testplattform entwickelten die Studenten Prototypen für fünf Anwendungen und untersuchten ihre Eignung für das autonome Fahren:

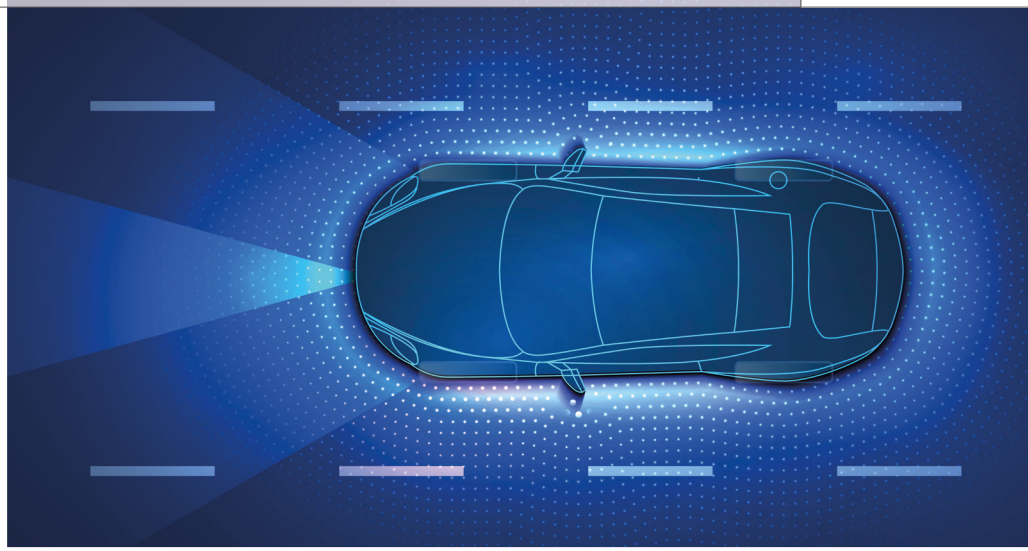
1. Erstellen eines benutzerspezifischen Komponentenpakets
2. Entwickeln einer neuronalen Netzwerk-Python-Komponente
3. Erkennen von Fußgängern
4. Aufzeichnen und Wiedergeben von Echtzeitszenarien
5. Kollisionswarnung mit SVM-Klassifikator

Anwendung 1: Erstellung eines benutzerspezifischen Komponentenpakets

Die Studenten nutzten die grafische Entwicklungsumgebung RTMaps Studio und die RTMaps Component Library sowie integrierten C/C++- und Python-Code, um ein Komponentenpaket zu erstellen. Für den Import des Komponentenpakets in den RTMaps-Projektarbeitsbereich kam RTMaps SDK zum Einsatz. RTMaps Studio stellt Anwendungen durch verbundene Komponenten dar, die von der RTMaps Component Library bereitgestellt werden. „Mit Studio lassen sich komplexe, modulare Anwendungen komfortabel einrichten", sagt Ingenieurstudent Sreeram Venkitachalam. „Komponenten können einfach aus der Bibliothek in den Arbeitsbereich von Studio gezogen werden. Die Komponenten werden für die Kommunikation, die Anbindung von Sensoren, den Aufbau eines Algorithmus und den Anschluss der Aktoren verwendet." Abbildung 1 veranschaulicht ein benutzerdefiniertes Komponentenpaket, das in RTMaps Studio importiert wurde. Mit dem RandomCANFrameGenerator-Paket wurden sowohl standardisierte als auch erweiterte CAN-Frames generiert. Die eingehenden CAN-Daten wurden so gefiltert, dass der Daten-Viewer in RTMaps nur die Standard-Frames anzeigt. Es stehen zwei Optionen zur Verfügung, um die Geschwindigkeit der CAN-Frame-Generierung zu steuern: Die Option „speed up" verdoppelt die aktuelle Geschwindigkeit der CAN-Generierung, wohingegen die Option „speed down" diese halbiert.

Anwendung 2: Entwickeln einer neuronalen Netzwerk-Python-Komponente

Als nächste Anwendung entwickelten



die Studenten eine neuronale Netzwerk-Python-Komponente und klassifizierten das Eingangsbild. Dazu wurde die Python-Komponente des RTMaps-Embedded-Pakets verwendet. Diese enthält einen Editor mit Syntaxfärbung, der die Benutzer bei der Entwicklung von Python-Skripten unterstützt. Der Editor steht per Kontextmenü für jedes Komponentenmodul direkt zur Verfügung. Der Python-Block von RTMaps generiert eine Klasse namens 'RTMaps_Python', mit der Eingaben reaktiv oder periodisch aufgerufen werden können. Die Kernfunktion der Klasse RTMaps_Python arbeitet als Endlosschleife, die das Hauptprogramm ausführt. Mit der Python-Komponente von RTMaps Embedded konnten die Studenten zwei Beispielprojekte abschließen: Fahrzeugerkennung und Verkehrszeichenklassifizierung.

Fahrzeugerkennung

Um Fahrzeuge zu erkennen, wurde mit einem fünfschichtigen neuronalen Faltungsnetzwerk (Convolutional Neural Network, CNN) und Pooling in jeder Schicht ein Bild implementiert. Das Netzwerk nutzte zwei vollständig verbundene Schichten zusammen mit einer Ausfallschicht, um eine Überlastung des Netzwerks zu verhindern. Das Modell wurde mit einem Regressionsmodell trainiert und mit einem Adam-Optimierer verfeinert. Die Struktur des neuronalen Netzwerks wurde dem RTMaps-Python-Block hinzugefügt, und die trainierte Gewichtung für den Block wurde dem Netzwerk als gespeicherte Daten zugeführt. Das Netzwerk im Python-Block prüft, ob ein Bild ein Fahrzeug enthält. Die RTMaps-Anwendung wird an BlueBox gesendet und dort ausgeführt. Die RTMaps-Python-Bibliotheken in BlueBox wurden so überarbeitet, dass eine TensorFlow Deep Learning Library (tflearn) zum Aufbau der CNN-Struktur eingebunden werden konnte.

Verkehrszeichenklassifizierung

In ihrem zweiten Beispielprojekt konnten die Studenten deutsche Verkehrszeichen mit einem TensorFlow-Modell klassifizieren. Mit dem Modell identifizierten sie die Verkehrszeichen, ordneten jedes einer von 43 Klassen zu und zeigten die Bedeutung des erkannten Verkehrszeichens im Bild an. Für den Lernprozess nutzte das Klassifizierungsmodell ca. 39.000 Bilder. Das TensorFlow-Modell wurde für Testzwecke neu erstellt, und die mit dem Trainingsmodell gespeicherten Modellgewichtungen wurden hinzugefügt, um die Klasse des Eingangsbildes vorherzusagen. Die gesamte Teststruktur wurde in der RTMaps-Python-Komponente entworfen. Das trainierte Modell und die aufgenommenen Bilder wurden an die BlueBox gesendet. Anhand des im Bild vorhandenen Zeichens bekam die Ausgabe ein Label mit der Bedeutung des Verkehrszeichens.

> „Studio von RTMaps bietet uns eine einfache Möglichkeit, komplexe modulare Anwendungen aufzubauen."
>
> *Sreeram Venkitachalam, Ingenieurstudent an der IUPUI*

Anwendung 3: Erkennen von Fußgängern

In ihrer nächsten Anwendung gelang es den Studenten, Fußgänger durch die Erfassung von Echtzeit-Frames mit einer Axis-IP-Kamera (IPCAM) zu erkennen. Das Fußgängererkennungs- >>

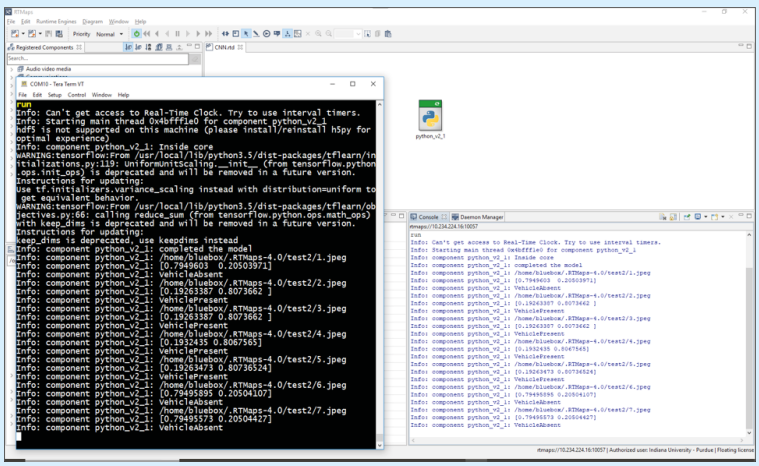


Abbildung 3: Ergebnisdarstellung der Fahrzeugerkennung basierend auf einem neuronalen Faltungsnetzwerk (CNN) unter Verwendung der BlueBox.

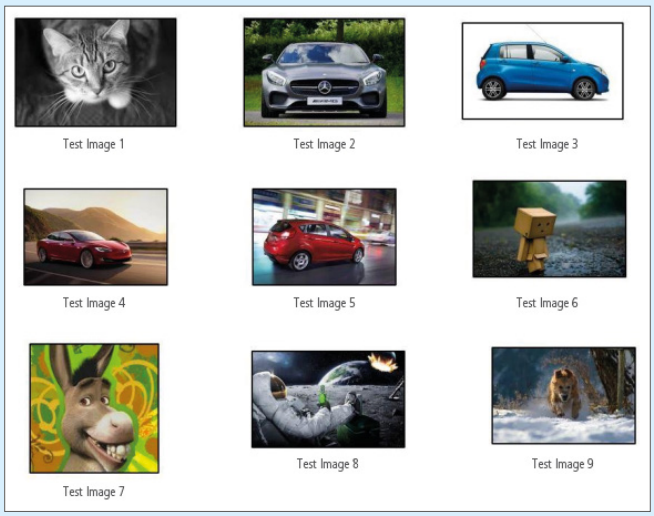


Abbildung 4: Eingangsbilder zum Testen der Fahrzeugerkennung.

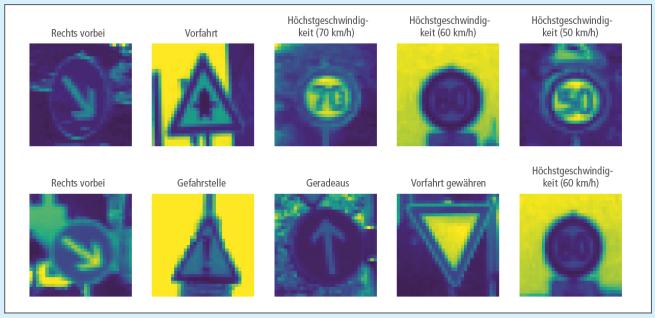


Abbildung 5: Deutsche Verkehrszeichen, die mit einem TensorFlow-Modell klassifiziert wurden.

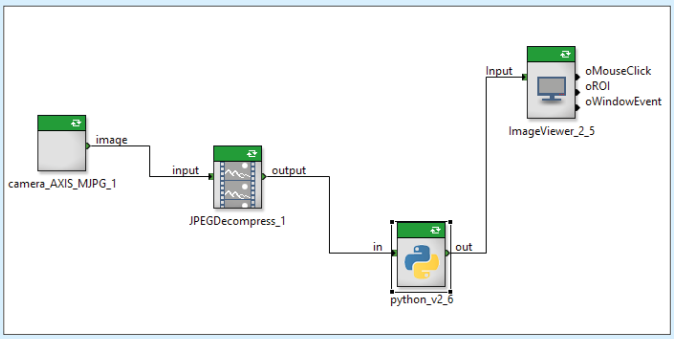


Abbildung 6: Diagramm der IPCAM-basierten Fußgängererkennung.

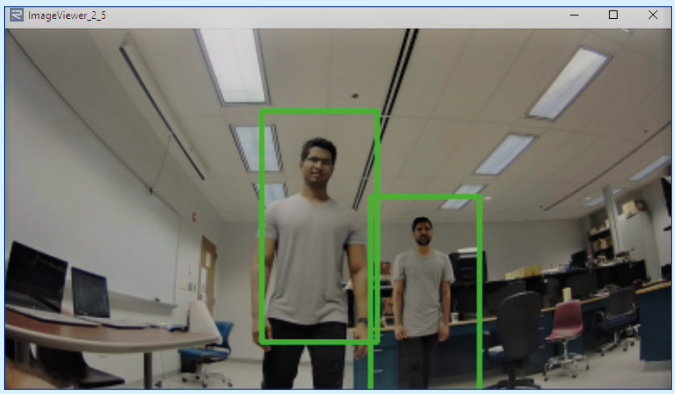


Abbildung 7: Begrenzungsrahmen werden um erkannte Fußgänger gelegt.

> „RTMaps Embedded ist geeignet, um die Multisensor-Herausforderungen zu meistern. Das Tool schafft einen effizienten und intuitiven Rahmen für die schnelle und robuste Entwicklung von Fahrerassistenzsystemen, Fahrzeugen und Robotik."
>
> *Professor Mohamed El-Sharkawy, Purdue School of Engineering and Technology*

modell in der Python-Anwendung von RTMaps identifiziert Personen im Eingabebild und zeichnet für jede einen Begrenzungsrahmen (Abbildung 7). Als vortrainierter Detektor wurden ein HOG (Histogram of oriented Gradients) und eine linear unterstützte Vektormaschine (SVM) verwendet. Um zu verhindern, dass sich die Begrenzungsrahmen überschneiden, wurde die Non-Maxima-Unterdrückung (NMS) aktiviert. Das Fußgängererkennungsmodell wurde in das IPCAM integriert, die Eingangsbilder wurden aus dem IPCAM in die Python-Anwendung eingespeist. Die erkannten Fußgänger und zugehörigen Begrenzungsrahmen wurden anschließend im Image Viewer von RTMaps dargestellt.

Anwendung 4: Aufzeichnen und Wiedergeben von Echtzeitszenarien

In ihrer vierten Anwendung konnten die Studenten zeigen, wie leistungsstark RTMaps hinsichtlich der Aufzeichnung und Wiedergabe von Echtzeitdaten ist. Die erfassten Daten wurden als REC-Datei gespeichert, die die Studenten wiedergeben konnten. In diesem Beispiel wurde das IPCAM in die LS2084A der BlueBox integriert, um die Bilder aufzunehmen und als REC-Dateien im IPCAM-Ordner zu speichern.

Anwendung 5: Kollisionswarnung mit SVM-Klassifikator

Mit ihrer abschließenden Anwendung machten sich die Studenten daran, ein vorausschauendes Kollisionswarnsystem zu implementieren, das ein zukunftsweisendes Fahrzeugradarmodell und einen Klassifizierungsalgorithmus enthält. Die Ergebnisse des Radarmodells waren die Geschwindigkeit/Beschleunigung und der Abstand zwischen den Fahrzeugen. Diese Outputs wurden als Inputs für die lineare Regression und den SVM (Support Vektor

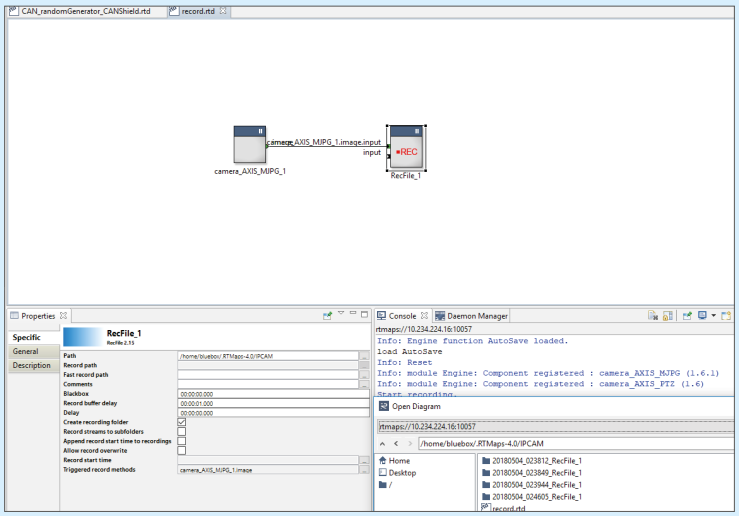


Abbildung 8: Beispiel für eine BlueBox-Aufzeichnung.

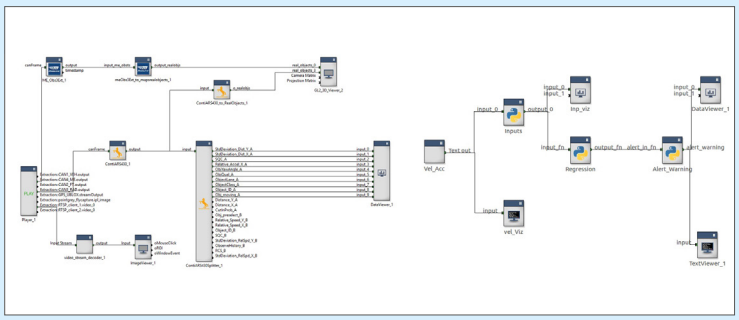


Abbildung 9: Diagramm einer Kollisionswarnung mit SVM-Klassifikator.

Machine)-Klassifikator verwendet, um den Warnbereich vorherzusagen – also den Bereich, in dem das Ego-Fahrzeug mit dem vorausfahrenden Fahrzeug kollidieren könnte.

Fazit und Ausblick

Die IUPUI-Ingenieurstudenten konnten mit der eingebetteten Plattform RTMaps und der NXP BlueBox verschiedene Anwendungsprojekte für autonomes Fahren realisieren. Sie entwickelten und testeten fünf unterschiedliche Anwendungsbeispiele, um zu bestätigen, dass Intempora RTMaps 4.5.0 mit der Berechnungsplattform BlueBox 2.0 integriert werden kann. Während das Datenvolumen und die Komplexität der ADAS-Algorithmen zunahmen, profitierten die Studenten von der Rechenleistung von RTMaps und BlueBox. So konnten sie sicherstellen, dass das autonome Fahrzeug die Verkehrsumgebung und die Fahrbedingungen schnell erkennt und darauf reagiert. ■

Mit freundlicher Genehmigung der Indiana University – Purdue University Indianapolis

Mitwirkende der Purdue School of Engineering and Technology:

Mohamed El-Sharkawy
Professor
Fulbright-Stipendiat
Director of the Internet of Things (IoT) Collaboratory

Akash Gaikwad
Doktorand
IoT Collaboratory
Department of Electrical and Computer Engineering

Surya Kollazhi Manghat
Doktorandin
IoT Collaboratory
Department of Electrical and Computer Engineering

Sreeram Venkitachalam
Doktorand
IoT Collaboratory
Department of Electrical and Computer Engineering, derzeit Test- und Verifikationsingenieur bei Aptiv

Niranjan Ravi
Doktorand
IoT Collaboratory
Department of Electrical and Computer Engineering

Sree Bala Shruthi Bhamidi
Doktorandin
IoT Collaboratory
Department of Electrical and Computer Engineering

Dewant Katare
Doktorand
IoT Collaboratory
Department of Electrical and Computer Engineering