

Machine Learning und Deep Learning ist durch erfolgreiche Anwendungen (ChatGPT, Alexa, Tesla, usw.) in aller Munde. Python bietet dafür eine hervorragende Programmiersprache mit unzähligen Drittanbieter-Bibliotheken.

### Ihr Nutzen

Sie lernen in diesem Workshop Grundlagen von Machine und Deep Learning und wie Sie eigene Anwendungen basierend u.A. auf Python, TensorFlow, OpenCV und PyTorch umsetzen können.

### Voraussetzungen

Python Object Oriented Scripting~5579

### Preis pro Teilnehmer

EUR 2950,- exklusive der gesetzlichen MwSt.

### Seminardauer

5 Tag(e)/Day(s)

### Hinweise

Sie erhalten im Seminar einen aktuellen Raspberry PI zur Umsetzung der Übungen, den Sie nach dem Seminar mit nach Hause nehmen können.

### Seminarinhalte

#### 1. Tag

\* Einführung in Künstliche Intelligenz (KI)

\* Machine Learning (ML)

- Einsatz in der Praxis, Typische Aufgaben und Lösungen

\* Deep Learning (DL)

- Einsatz in der Praxis, Typische Aufgaben und Lösungen  
- Neuronale Netze

\* Unterschiede und Anwendung DL / ML

\* Python als Programmiersprache für Machine Learning

- Warum Python? Vorteile der Sprache für KI  
- Voraussetzungen Hardware/Software

\* Virtuelle Umgebungen in Python

- Einrichten und Konfiguration mit den entsprechenden Kommandos  
- Modulkonfiguration in Virtuellen Python Umgebungen  
- Typische Konfiguration für Machine Learning (Modulübersicht)

#### 2. Tag

\* Module im Detail

- Python Module NumPy, SciPy, Pandas mit Funktionsbeispielen

\* TensorFlow/TensorFlow Light Übersicht der Funktionalität und Leistungsfähigkeit

- Grundlagen und Architektur  
- Beispieldaten in der Praxis  
- Historie, Funktionsweise und Anforderungen an Soft und Hardware

\* TensorFlow Komponenten

- Tensor, Vektoren, Matrizen  
- Graphs

\* Laden von Daten in TensorFlow, Varianten

- Laden in den Speicher  
- TensorFlow-Pipeline

\* Teachable Machine

- Überblick  
- Nutzen für eigene Entwicklungen  
- Modell trainieren + testen

Version: N/A

#### 3. Tag

\* TensorFlow Lite Advanced

- Eigene Modelle mit Model Maker trainieren  
- Performance Optimierung  
- TPU (Tensor Processing Unit) Überblick  
- Google Coral USB-Beschleuniger (Features im Überblick)

\* Keras

- Einsatzgebiet und Vergleich mit TensorFlow  
- MNIST-Datensatz / Standard für Mustererkennung  
- Trainieren des neuronalen Netzwerks

\* Workshop: Korrektes zuzuordnen von handschriftlichen Ziffern

#### 4. Tag

\* OpenCV

- Funktionsübersicht  
- Einsatz in Bilderkennung und Bearbeitung  
- Laden und darstellen von Bildern  
- Zugriff auf Pixel  
- Resizen von Bildern  
- Rotieren und Bildverbesserung  
- Objekterzeugung  
- Objekterkennung  
- Gestenerkennung  
- Gesichtserkennung

#### 5. Tag

\* PyTorch

- Funktionsübersicht  
- Vergleich mit TensorFlow  
- Dynamic Computational Graph (DAG)  
- Benutzerfreundlichkeit / Visualisierung ?  
- Modelldefinitionen  
- PyTorch Modelle optimieren  
- Modelle speichern und laden  
- Mit Daten in PyTorch arbeiten  
- (torch.utils.data.DataLoader und torch.utils.data.Dataset

\* Workshop mit vorbereiteten Frameworks, Modulen und APIs

