

# Handgestenerkennung mit einer Smart-Cam unter Verwendung eines neuronalen Netzes

## Projekt 1 / SoSe 2020

*In diesem Projekt wurde ein neuronales Netz auf einer Smart Camera implementiert, die damit 10 unterschiedliche Handgesten erkennen kann.*

### Hardware

Eine Smart Kamera hat einen Prozessor integriert und benötigt zur Bildauswertung keinen externen Computer, Bildschirm o.ä. Von der Firma Imago wurde uns freundlicherweise eine *Vision Sensor PV* zur Verfügung gestellt. Auf der Kamera läuft ein Linux als Betriebssystem. Die Kamera ist ein komplettes Modul mit Objektiv und digitalen I/O-Schnittstellen, sowie einer Ethernet-Anbindung. Zur Programmierung und zur Abfrage der Ergebnisdaten kann mithilfe einer SSH Verbindung auf die Kamera zugegriffen werden.



### Neuronale Netze

Ein künstliches neuronales Netzwerk ist vom Konzept her an biologische Vorbilder (Gehirne) angelehnt; es nutzt künstliche Neuronen / Verbindungen zur Informationsverarbeitung. Das hier verwendete Neuronale Netz gehört zu den sogenannten Convolutional Neural Networks (CNN), die speziell auf Anwendungen in der Bildverarbeitung ausgelegt sind. So ein neuronales Netzwerk kann man mit Bildern, z.B. von Handgesten, so trainieren, dass diese Gesten erkannt werden.

### Team



Jan Sievers, Franka Ludig, Lukas Reißer, Julian Odloschinski

### Programmierung

Als Entwicklungsumgebung wurde Visual Studio eingesetzt, an das die Kamera angebunden werden kann. Das Programm selbst wurde in C++ geschrieben und für den Kameraprozessor kompiliert. In dieses Programm wurde ein fertiges Neuronales Netz eingebunden, das frei im Netz verfügbar ist und

bereits mit 20 000 unterschiedlichen Handgesten-Bilder trainiert wurde. Zusätzlich wurde eine Belichtungssteuerung für die Kamera und eine einfache Bildvorverarbeitung realisiert. Dafür und für die Bildaufnahme wurde auf Funktionen der OpenCV-Bibliothek zurückgegriffen.

Eines der Probleme im Projekt war, dass dieses Netz die Tensorflow-Bibliothek benutzt, die aber nur als 64-Bit-Code vorliegt, während das Kamera-OS ein 32-Bit-System ist. Durch Konvertierung konnte dieses Hindernis letztlich überwunden werden.

## Ergebnis

Das Ziel wurde größtenteils erreicht: Das neuronale Netz läuft auf der Kamera auch ohne externe Unterstützung, allerdings ist die Ausgabe dessen noch in der Konsole. So sieht eine typische Ausgabe aus:

```
C
Thumbs down: 0.01%
Palm (Horizontal): 0.00%
L: 0.00%
Fist (Horizontal): 0.00%
Fist (Vertical): 0.00%
Thumbs up: 0.00%
Index: 0.00%
OK: 0.02%
Palm (Vertical): 0.00%
C: 99.95%
```

**Out[48]:** <matplotlib.image.AxesImage at 0x7f9cbfaa6080>



Die Vorgehensweise wurde vom Team in einem Tutorial zusammen gefasst.